Intel® 835 Chipset: 82835 Graphics and Memory Controller Hub (GMCH)

Datasheet

December 2003

Order Number: 300401-001
December 2003

Contents

Figures

Tables

Revision History

Date	Revision	Description
December 2003	001	Initial release

1.0 Introduction

1.1 Product Features

The Intel® 835 chipset includes many of the same core features as the Intel® 830M chipset. The following section, "Intel® 830M Chipset Core Functionality" lists the core features of the Intel 830M chipset that are also found in the Intel 835 chipset. For a complete description of all features of the Intel 830M chipset, refer to the Intel 830 chipset datasheet located at http://developer.intel.com/design/chipsets/datashts/298338.htm.

1.1.1 Intel® 830M Chipset Core Functionality

The core features of the Intel 830M chipset include the following:

- Supports PC133 SDRAM.
- The DVO interface acts as a master only. There is only one available DVO port.
- The DVO interface meets the requirements found in document FM-2035.
- Supports 133 MHz processor system bus.
- Supports the video overlay.
- Supports the SMBUS.

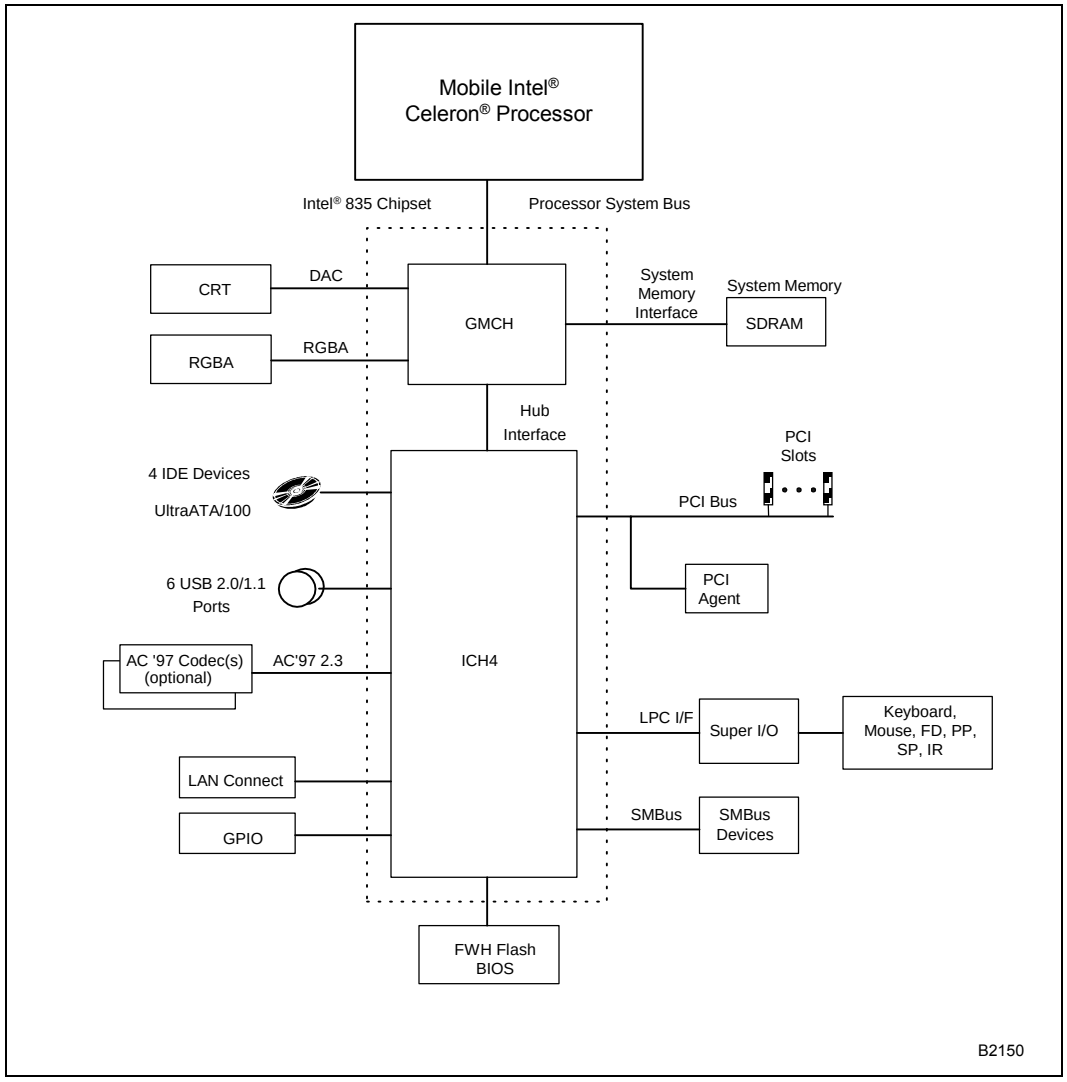
1.1.2 Intel® 835 Chipset Additional Functionality

The Intel 835 chipset includes the following additional features:

- The TV pixel clock input could use either the 3.3 V DREFCLK or the 1.5 V RGBA_CLKIN clock.
- Supports +-31 RGBA output clock window for placement of falling edge of VSYNC relative to HSYNC for detecting odd/even field.
- If VSYNC rising edge is detected within +-31 clocks of HSYNC, then an odd field is detected; otherwise the field is even.
- Supports interlaced timing using Field ID from the timing generator.
- Provides 8 bits of per-pixel alpha values on the RGBA interface accomplished by four outputs double-pumped as described in Table 88.
- Supports TR-B15 Operational Guidelines for Digital Satellite Broadcasting (detailed Implementation guideline for receiver).
- Supports pixel duplication and line replication for converting 960x540 and 960x1080 to 1920x1080i for ARIB standard.
- Supports the resolutions outlined in Table 89.
- Supports active and blank timing for integrated circuit suppliers.
- Outputs final pixel color and corresponding alpha value.
- The overlay plane is displayed using color keying only. In order to display the overlay plane, a color must be used to key the overlay image into the display.

- Supports ICH4 desktop.

Figure 1. Intel® 835 Chipset Interface Block Diagram



2.0 Design Specifications

This section provides the external design specifications for the Intel® 835 chipset GMCH.

2.1 Document References

- *Mobile Intel® Celeron® Processor (0.18µ) in Micro-FCBGA and Micro-FCPGA Packages Datasheet* (298514):
  Contact http://developer.intel.com/design/mobile/datashts/298514.htm
- *Mobile Intel® Celeron® Processor (0.13µ) in Micro-FCBGA and Micro-FCPGA Packages Datasheet* (298517):
  Contact http://developer.intel.com/design/mobile/datashts/298517.htm
- *PCI Local bus Specification 2.2*: Contact http://www.pcisig.com
- *Intel® 82801 DB I/O Controller Hub 4 (ICH4) Datasheet* (290716):
  Contact http://developer.intel.com/design/chipsets/datashts/290744.htm
- *Intel® Chipset Family: 82830 Graphics and Memory Controller Hub (GMCH-M) Datasheet* (298338):
  Contact http://developer.intel.com/design/chipsets/datashts/298338.htm
- *Intel® 830 Chipset Family Design Guide* (298339):
  Contact http://developer.intel.com/design/chipsets/designex/298339.htm
- *Intel® 835 Chipset Family Design Guide*
- *Advanced Configuration and Power Management (ACPI) Specification*:
  Contact http://www.acpi.info/spec.htm
- *Advanced Power Management (APM) Specification 1.2*:
  Contact http://www.microsoft.com/hwdev/busbios/amp_12.htm
- *Write Combing Memory Implementation Guideline*:
  Contact http://developer.intel.com/design/PentiumII/applnots/244422.htm
- *IA-32 Intel® Architecture Software Developer Manual Volume 3: System Programming Guide*:
  Contact http://developer.intel.com/design/Pentium4/manuals/245472.htm
- *Intel Graphics Software PC 10.0 Product Requirements:* Contact your Intel Field Representative.

2.2 Terminology

82835 Graphics and Memory Controller Hub (GMCH)	The Intel® 835 Graphics and Memory Controller Hub component that contains the CPU interface, system SDRAM controller, and Integrated Graphics Device (IGD). It communicates with the ICH4 over a proprietary interconnect called the hub interface.
Internal Graphics Device	PCI device #2 of the Intel 835 Graphics and Memory Controller Hub (GMCH) component, which implements the Intel Graphics solution. In this document, PCI Device #2 is referred to as the Intel Graphics Device.
82801 DB I/O Controller Hub (ICH4)	The ICH4 is connected to the GMCH through a proprietary interconnect called the hub interface. This is the I/O Controller Hub or ICH component that contains the primary PCI interface, LPC interface, USB2.0, ATA-133 and other I/O functions.
Hub Interface	The proprietary interconnect between the GMCH and the ICH4. In this document, hub interface cycles originating from or destined for the ICH4 are generally referred to as hub interface cycles. Hub cycles originating from or destined for the primary PCI interface on the ICH4 are sometimes referred to as Hub Interface/PCI cycles.
RGBA Port	Red/Green/Blue/Alpha. Refers to the Intel 835 chipset's digital display channels. The Intel 835 chipset has one dedicated RGBA port.
Primary PCI	The primary physical PCI (PCI0) bus that is driven directly by the ICH4 component. It supports a 3.3-V interface and is 5.0 V tolerant, 33 MHz PCI 2.2 compliant components. Interaction between PCI0 and GMCH occurs over the hub interface. Note that even though the Primary PCI bus is referred to as PCI0, it is not PCI Bus #0 from a configuration standpoint.
Secondary PCI	The secondary physical PCI (PCI1) interface that is a subset of the AGP* bus driven directly by the GMCH. It supports a subset of 1.5 V, 66 MHz PCI 2.2 compliant components. Note that even though the Secondary PCI bus is referred to as PCI1, it may not be configured as PCI Bus #1. Note: AGP capabilities are not functional in the Intel 835 chipset.
UMA	Unified Memory Architecture. Graphics memory for the IGD that is located in system memory.
IGD	Integrated Graphics Device. The graphics device that is internal to the GMCH.
DVMT	Direct Video Memory Technology
BLT	Block Transfers of Data

2.3 System Architecture

The Intel 835 chipset is a highly integrated hub that provides the CPU interface to a Mobile Intel Celeron® processor, the SDRAM system memory interface, a hub interface to the 82801DB I/O Controller Hub (ICH4), and is optimized for Mobile Intel Celeron processor configurations at 133 MHz processor system bus (PSB).

2.3.1 Intel® 835 Chipset

The Intel 835 chipset has integrated graphics capabilities. Its dedicated multimedia engines deliver high-performance 3D, 2D, video, and display capabilities.

- 1.25-V AGTL host bus supports 32-bit host addressing

- System SDRAM supports PC133 (LVTTL) SDRAM
- Supports up to 512 Mbytes of system SDRAM
- Hub interface to ICH4
- Integrated graphics capabilities, including 3D rendering acceleration and 2D hardware acceleration
- Integrated 165-MHz RAMDAC
- A variety of display device protocols (TV and DVI) are supported through the RGBA port connected to external devices

2.4 Host Interface

The Intel 835 chipset family is optimized for Mobile Intel Celeron processors. The Intel 835 chipset supports a processor system bus frequency of 133 MHz using 1.25-V AGTL signaling. Dual-ended termination AGTL is supported for single-processor configurations. It supports 32-bit host addressing, decoding up to 4 Gbytes of the CPU's memory address space. Host initiated I/O cycles are decoded to PCI1, hub interface, or GMCH configuration space. Host initiated memory cycles are decoded to PCI1, hub interface, or system SDRAM. GMCH accesses to graphics memory and PCI1 device accesses to non-cacheable system memory are not snooped on the host bus. Memory accesses initiated from PCI1 using PCI semantics and from hub interface to system SDRAM are snooped on the host bus.

2.5 System Memory Interface

The Intel 835 chipset integrates a system memory SDRAM controller with a 64-bit wide interface. The Intel 835 chipset supports Single Data Rate (SDR) SDRAM for system memory. Consequently, the GMCH system memory buffers support LVTTL signal interfaces.

Configured for SDRAM, the GMCH memory interface includes support for the following:

- Up to 512 Mbytes of 133-MHz SDR SDRAM using 256-Mbit technology
- PC133 DIMMs
- Maximum of two DIMMs, single-sided and/or double-sided
- The Intel 835 chipset only supports four bank memory technologies
- Four integrated clock buffers

2.6 Internal Graphics Device Introduction

The Intel 835 chipset IGD provides a highly integrated graphics accelerator delivering high-performance 3D, 2D, and video capabilities. The GMCH contains an extensive set of instructions for 3D operations, block transfers of data (BLT) and stretch BLT operations, motion compensation, overlay, and display control. The GMCH supports a UMA architecture using a DVMT configuration.

High-bandwidth access to data is provided through the system memory ports. The GMCH uses tiling architecture to increase memory efficiency and maximize effective rendering bandwidth. The GMCH uses Intel's Direct Memory Execution model to fetch textures from system memory at 1.0 Gbyte/s. The GMCH includes a cache controller to avoid frequent memory fetches of recently used texture data.

The GMCH also provides 2D hardware acceleration for BLTs. The BLT engine provides the ability to copy a source block of data to a destination and perform raster operations (e.g., ROP1, ROP2, and ROP3) on the data using a pattern, and/or another destination. Performing these common tasks in hardware reduces CPU load, and improves performance.

2.7 Internal Graphics Display Interface

The Intel 835 chipset has two display ports, one analog and one digital. This provides support for a progressive scan analog monitor, and a dedicated RGBA port. Each port can transmit data according to one or more protocols. The RGBA port is connected to an external device that converts one protocol to another. Examples of this are TV encoders, external DACs, and TMDS transmitters. Each display port has control signals that may be used to control, configure and/or determine the capabilities of an external device. The data that is sent through the display port comes from pipe A.

2.7.1 Analog Display Port

The Intel 835 chipset has an integrated 165-MHz RAMDAC that can directly drive a progressive scan analog monitor up to a resolution of 1600x1200 pixels.

2.7.2 RGBA Interface

The Intel 835 chipset provides a digital display channel that is capable of driving a pixel clock up to 81.23 MHz.

3.0 Signal Description

This section provides a detailed description of the Intel® 835 chipset GMCH signals. The signals are arranged in functional groups according to their associated interface.

The "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name the signal is asserted when at the high voltage level.

When not otherwise specified, "set" refers to changing a bit to its asserted state (a logical 1). Clear refers to changing a bit to its negated state (a logical 0).

The following notations are used to describe the signal type:

I	Input pin
O	Output pin
I/O	Bidirectional Input/Output pin
s/t/s	Sustained tristate. This pin is driven to its inactive state prior to tri-stating.
as/t/s	Active Sustained Tristate. This applies to some of the hub interface signals. This pin is weakly driven to its last driven value.

The signal description also includes the type of buffer used for the particular signal:

AGTL	Open Drain 1.25-V AGTL interface signal. Refer to the AGTL I/O Specification for complete details. AGTL signals are "inverted bus" style where a low voltage represents a logical "1".
LVTTL	Low Voltage TTL compatible signals. These are also 3.3-V outputs.
CMOS	CMOS buffers.

Note that the CPU address and data bus signals are logically inverted signals. In other words, the actual values are inverted from what appears on the CPU bus. This must be taken into account and the addresses and data bus signals must be inverted inside the GMCH. All CPU control signals follow normal convention. A "0" indicates an active level (low voltage) if the signal is followed by the # symbol and a "1" indicates an active level (high voltage) if the signal has no # suffix.

Table 1 shows the Vtt/Vdd and Vref levels for the various interfaces.

Table 1. Signal Voltage Levels

Interface	Vtt/Vdd (nominal)	Vref
AGTL	1.25 V	2/3 * Vtt
RGBA 1.5 V	1.5 V	0.5 * Vdd
LVTTL	3.3 V	Vddq * 0.5
RSL [Reserved]	1.8 V	1.4 V
Hub Interface	1.8 V	0.5 * Vdd

3.1 Common Signals for the Intel® 835 Chipset

This section describes the common signals that apply to the Intel 835 chipset. The Intel 835 chipset common signals consist of host interface signals, system memory signals, hub interface signals, and clock and reset signals.

3.1.1 Host Interface Signals

Table 2. Host Interface Signal Descriptions (Sheet 1 of 2)

Signal Name	Type	Description
H_CPU RSTB	O AGTL	**CPU Reset**. The H_CPU RSTB signal is an output from the GMCH. The GMCH asserts H_CPU RSTB while RSTB (PCIRST# from ICH4) is asserted and for approximately 1 ms after RSTB is deasserted. The H_CPU RSTB allows the CPUs to begin execution in a known state. Note that the ICH4 must provide CPU strap set-up and hold times around H_CPU RSTB. This requires strict synchronization between GMCH H_CPU RSTB deassertion and ICH4 driving the straps.
H_A [31:3]	I/O AGTL	**Host Address Bus**: H_A [31:3] connect to the CPU address bus. During CPU cycles the H_A [31:3] are inputs. The GMCH drives H_A [31:3] during snoop cycles on behalf of hub interface and secondary PCI initiators.
H_D [63:0]	I/O AGTL	**Host Data**: These signals are connected to the CPU data bus. Note that the data signals are inverted on the CPU bus.
H_ADSB	I/O AGTL	**Address Strobe**: The CPU bus owner asserts H_ADSB to indicate the first of two cycles of a request phase.
H_BNRB	I/O AGTL	**Block Next Request**: Used to block the current request bus owner from issuing a new request. This signal is used to dynamically control the CPU bus pipeline depth.
H_BPRIB	O AGTL	**Priority Agent Bus Request**: The GMCH is the only priority agent on the CPU bus. It asserts this signal to obtain the ownership of the address bus. This signal has priority over symmetric bus requests and causes the current symmetric owner to stop issuing new transactions unless the H_LOCKB signal was asserted.
H_DBSYB	I/O AGTL	**Data Bus Busy**: Used by the data bus owner to hold the data bus for transfers requiring more than one cycle.
H_DEFERB	O AGTL	**Defer**: GMCH generates a deferred response as defined by the rules of the GMCH's dynamic defer policy. The GMCH also uses the H_DEFERB signal to indicate a CPU retry response.
H_DRDYB	I/O AGTL	**Data Ready**: Asserted for each cycle that data is transferred.
H_HITB	I/O AGTL	**Hit**: Indicates that a caching agent holds an unmodified version of the requested line. Also, driven in conjunction with H_HITMB by the target to extend the snoop window.
H_HITMB	I/O AGTL	**Hit Modified**: Indicates that a caching agent holds a modified version of the requested line and that this agent assumes responsibility for providing the line. Also, driven in conjunction with H_HITB to extend the snoop window.
H_LOCKB	I AGTL	**Host Lock**: All CPU bus cycles sampled with the assertion of H_LOCKB and H_ADSB, until the negation of H_LOCKB must be atomic, i.e. no hub interface or PCI snoopable access to SDRAM is allowed when H_LOCKB is asserted by the CPU.

Table 2. Host Interface Signal Descriptions (Sheet 2 of 2)

Signal Name	Type	Description
H_REQB [4:0]	I/O AGTL	**Host Request Command**: Asserted during both clocks of request phase. In the first clock, the signals define the transaction type to a level of detail that is sufficient to begin a snoop request. In the second clock, the signals carry additional information to define the complete transaction type. The transactions supported by the GMCH host bridge are defined in Section 2.4 of this document.
H_TRDYB	I/O AGTL	**Host Target Ready**: Indicates that the target of the CPU transaction is able to enter the data transfer phase.
H_RSB [2:0]	I/O AGTL	**Response Signals**: Indicates type of response according to the following table: H_RSB [2:0] Response type 000 Idle state 001 Retry response 010 Deferred response 011 Reserved (not driven by GMCH) 100 Hard failure (not driven by GMCH) 101 No data response 110 Implicit writeback 111 Normal data response
H_RCOMP	I/O	**GTL Compensation**: Used to calibrate the GTL interface buffers to match the board. This pin should be connected to an 80-Ω simple resistor to ground.

NOTE: Total pins for this section: 113

3.1.2 System Memory Interface

Table 3. System Memory Interface Signal Descriptions (Sheet 1 of 2)

Signal Name	Type	Description
SMA [12:0]	O LVTTL	**Memory Address**: SMA [12:0] is used to provide the multiplexed row and column address to SDRAM.
SMBA [1:0]	O LVTTL	**Memory Bank Address**: These signals define the banks that are selected within each SDRAM row. The SMA and SMBA signals combine to address every possible location within a SDRAM device.
SMD [63:0]	I/O LVTTL	**Memory Data**: These signals are used to interface to the SDRAM data bus.
SMDQM [7:0]	O LVTTL	**Input/Output Data Mask**: These pins act as synchronized output enables during read cycles and as byte enables during write cycles.
SMCS [3:0]	O LVTTL	**Chip Select**: For the memory rows configured with SDRAM, these pins perform the function of selecting the particular SDRAM components during the active state. Note: There is one SMCS per SDRAM row. These signals can be toggled on every rising system memory clock edge.
SMRAS	O LVTTL	**SDRAM Row Address Strobe**: A table of the SDRAM commands supported by the Intel® 835 chipset is given in the SDRAM section. SMRAS may be heavily loaded and requires 2 SDRAM clock cycles for set-up time to the SDRAMs.
SMCAS	O LVTTL	**SDRAM Column Address Strobe**: A table of the SDRAM commands supported by the Intel 835 chipset is given in the SDRAM section. SMCAS may be heavily loaded and requires 2 SDRAM clock cycles for set-up time to the SDRAMs.

Table 3. System Memory Interface Signal Descriptions (Sheet 2 of 2)

Signal Name	Type	Description
SMWE	O LVTTL	**Write Enable Signal**: SMWE is asserted during writes to SDRAM. Refer to truth table of the SDRAM commands supported by the Intel 835 chipset, given in the SDRAM section. SMWE may be heavily loaded and requires 2 SDRAM clock cycles for set-up time to the SDRAMs.
SMCKE [3:0]	O LVTTL	**Clock Enable**: These signals are used to signal a self-refresh or power-down command to a SDRAM array when entering system suspend. SMCKE is also used to dynamically power down inactive SDRAM rows. There is one SMCKE per SDRAM row. These signals can be toggled on every rising SMCLK clock edge.
SMOCLK	O LVTTL	**System Memory Output Clock**: This signal delivers a synchronized clock to the SMRCLK pin.
SMRCLK	I LVTTL	**System Memory Return Clock**: This signal receives the synchronized clock from SMOCLK.
SMCLK [3:0]	O LVTTL	**System Memory Clock**: These signals deliver a synchronized clock to the SDRAMs.
SM RCOMP	I/O	**System Memory RCOMP**: Used to calibrate the system memory I/O buffers. This pin should be connected to a 27.5-Ω resistor tied to Vss.

NOTE: Total pins for system memory section: 105

3.1.3 Hub Interface Signals

Table 4. Hub Interface Signal Descriptions

Signal Name	Type	Description
HLD [10:0]	I/O (as/t/s) CMOS	**HLD [10:0]** Hub interface Signals. Signals used for the hub interface.
HLSTRB; HLSTRBB	I/O (as/t/s) CMOS	**HLSTRB; HLSTRBB** Hub interface Strobe/Complement. The two differential strobe signals used to transmit or receive packet data.
HL RCOMP	I/O	**HL_RCOMP** Hub interface compensation: Used to calibrate the hub I/O buffers. This signal has an external 55-Ω pull-down resistor.

NOTE: Total pins for this section: 14

3.1.4 Clocking and Reset

Table 5. Clocking and Reset Signal Descriptions

Signal Name	Type	Description
HTCLK; HTCLKB	I CMOS	**Host Clock In**: These pins receive a buffered host clock from the external clock synthesizer. This clock is used by all of the GMCH. The clock is also the reference clock for the graphics core PLL. This is a low voltage differential input.
GB CLKOUT	O LVTTL	**Hub Clock Reference Output**: This clock goes to the external Hub/PCI buffer.
GB CLKIN0	I LVTTL	**Hub Input Clock**: 66 MHz, 3.3-V input clock from external buffer hub-link interface.
GCLK; RCLK	O CMOS	**Graphics Memory Clock Out: (Reserved)** These signals should leave as NC (“Not Connected”).
DREF CLK	I LVTTL	**Display Clock Input:** This pin provides a 48-MHz input clock to the Display PLL that is used for 2D/Video/Flat Panel and DAC. **Pixel Clock Input:** This pin can provide a pixel clock input at a minimum of 13.5 MHz and a maximum of 81.23 MHz. This signal may be selected as a reference input for an external TV encoder.
RSTB	I LVTTL	**Reset In**: When asserted, this signal asynchronously resets the GMCH logic. This signal is connected to the PCIRST# output of the ICH4. The ICH4 drives this to 3.3 V. All PCI output and bi-directional signals are also tri-state compliant to PCI rev 2.2 specifications. This input should have a Schmidt trigger to avoid spurious resets. Note that this input needs to be 3.3-V tolerant.

NOTE: Total pins for clocks and resets section: 8

3.1.5 Reserved Signals

Table 6. Intel Reserved Signals

Signal Name	Type	Description
DQA [7:0]		Intel Reserved. Should be left NC (“Not Connected”)
DQB [7:0]		Intel Reserved. Should be left NC (“Not Connected”)
RQ [7:0]		Intel Reserved. Should be left NC (“Not Connected”)
CTM; CTM_B		Intel Reserved. Requires pull down.
CFM; CFM_B		Intel Reserved. Should be left NC (“Not Connected”)
CMD		Intel Reserved. Should be left NC (“Not Connected”)
SCK		Intel Reserved. Should be left NC (“Not Connected”)
SIO		Intel Reserved. Should be left NC (“Not Connected”)

NOTE: Total reserved pins: 31

3.2 Common Signals for Internal Graphics Implementation

An internal graphics device is available with the Intel 835 chipset. The following signals apply when the internal graphics device is chosen. The internal graphics device has support for a dedicated digital video port (RGBA) and an analog display. Refer to the *Intel® 835 Chipset Design Guide* regarding design recommendations for pins where no functionality is defined.

Table 7. Internal Graphics Status Signal Descriptions

Signal Name	Type	Description
G_Busy	OD	**G_Busy**: Output of the Intel 835 GMCH graphics controller to the ICH4, which indicates that certain internal graphics (IGD) activity is taking place. Assertion indicates to the ACPI software to not enter the C3 state. Assertion also causes a C3 exit if C3 was being entered, or was already entered when G_Busy went active. G_Busy will be inactive when the graphics controller is in any ACPI state other than D0. G_Busy must be pulled up to a voltage rail when turned off in the ACPI S3-S5 stages.

3.2.1 Dedicated Digital Video Port (RGBA)

Table 8. Dedicated Digital Video Port (RGBA) Signal Descriptions (Sheet 1 of 2)

Name	Type	Description
RGBA_CLKOUT0; RGBA_CLKOUT1	O 1.5 V	**RGBA Clock Output**: These pins provide a differential pair reference clock that can run up to 81.23 MHz. RGBA_CLKOUT0 (L29) corresponds to the primary clock out. RGBA_CLKOUT1 (L28) corresponds to the primary clock out. RGBA_CLKOUT0 and RGBA_CLKOUT1 need to be pulled up if the signals are NOT used when using internal graphics device.
RGBA_DATA [11:0]	O 1.5 V	**RGBA Data**: This data bus is used to drive 12-bit RGB data on each edge of RGBA_CLKOUT. This provides 24-bits of data per clock. RGBA_DATA [11:0] should be left as NC (not connected) if: the signals are NOT used when using internal graphics device.
RGBA_ALPHA [3:0]	O 1.5 V	**RGBA Alpha**: This data bus is used to drive 4-bit Alpha data on each edge of RGBA_CLKOUT. This provides 8-bits of data per clock. RGBA_ALPHA [3:0] should be left as NC (not connected) if the signals are NOT used when using internal graphics device.
RGBA_HSYNC	O 1.5 V	**Horizontal Sync**: HSYNC signal for the RGBA interface. The active polarity of the signal is programmable. RGBA_HYSNC should be left as NC (not connected) if the signal is NOT used when using internal graphics device.

Table 8. Dedicated Digital Video Port (RGBA) Signal Descriptions (Sheet 2 of 2)

Name	Type	Description
RGBA_VSYNC	O 1.5V	**Vertical Sync**: VSYNC signal for the RGBA interface. The active polarity of the signal is programmable. RGBA_VSYNC should be left as NC (not connected) if the signal is NOT used when using internal graphics device
RGBA_RCOMP	I/O	**RGBA Compensation**: Used to calibrate the RGBA I/O buffers. This signal needs to be pulled down to ground through an external resistor (resistance is based on board impedance).
RGBA_CLKIN	I 1.5 V	**RGBA Pixel Clock Input/Interrupt:** This input pin can be programmed to be either a reference input to a dot clock PLL (DPLL) or to be a second interrupt input. **RGBA Pixel Clock Input:** This pin can provide a pixel clock input at a minimum of 13.5 MHz and a maximum of 81.23 MHz. This signal may be selected as a reference input for an external TV encoder. **RGBA Interrupt:** When used as an interrupt input, the signal acts as an interrupt signal to the GMCH. RGBA_CLKIN needs to be pulled up if the signal is NOT used when using internal graphics device.

NOTE: Total pins for RGBA section: 22.

Table 9. RGBA Display Control Signals Descriptions

Pin Name	Type	Description
M_GPIO0	I/O 1.5 V	**M_GPIO0**: The specific function is I2C_CLK for the RGBA port. M_GPIO0 needs to be pulled up if: i) The signal is NOT used when using internal graphics device.
M_GPIO1	1.5 V	**M_GPIO1**: The specific function is I2C_DATA for the RGBA port. M_GPIO1 needs to be pulled up if: i) The signal is NOT used when using internal graphics device.
M_GPIO2	1.5 V	**M_GPIO2**: The specific function is DDC1_DATA for the RGBA port. M_GPIO2 needs to be pulled up if: i) The signal is NOT used when using internal graphics device.
M_GPIO3	1.5 V	**M_GPIO3**: The specific function is DDC1_CLK for the RGBA port. M_GPIO3 needs to be pulled up if: i) The signal is NOT used when using internal graphics device.

NOTE: Total pins for this section: 4.

3.2.2 Analog Display

Table 10. Analog Display Signal Descriptions

Pin Name	Type	Description
VSYNC	O LVTTL	**CRT Vertical Synchronization**: This signal is used as the vertical sync (polarity is programmable) or "VSYNC Interval." VSYNC should be left as NC ("Not Connected") if not implemented.
HSYNC	O LVTTL	**CRT Horizontal Synchronization**: This signal is used as the horizontal sync (polarity is programmable). HSYNC should be left as NC ("Not Connected") if not implemented.
RED	O Analog	**Red (Analog Video Output)**: This signal is a CRT Analog video output from the internal color palette DAC. The DAC is designed for a 37.5 Ω equivalent load on each pin (e.g., 75-Ω resistor on the board, in parallel with the 75-Ω CRT load). RED can be left as NC ("Not Connected") if not implemented.
GREEN	O Analog	**Green (Analog Video Output)**: This signal is a CRT analog video output from the internal color palette DAC. The DAC is designed for a 37.5 Ω equivalent load on each pin (e.g., 75-Ω resistor on the board, in parallel with the 75-Ω CRT load). GREEN can be left as NC ("Not Connected") if not implemented.
BLUE	O Analog	**Blue (Analog Video Output)**: This signal is a CRT Analog video output from the internal color palette DAC. The DAC is designed for a 37.5 Ω equivalent load on each pin (e.g., 75-Ω resistor on the board, in parallel with the 75-Ω CRT load). BLUE can be left as NC ("Not Connected") if not implemented.
IREF	I NA	**Resistor Set**: Set point resistor for the internal color palette DAC. A 255-W 1% resistor is required between IREF and VSSA. RSTB can be left as NC ("Not Connected") if not implemented.
REDW	O Analog	**Red**W**(Analog Output)**: This signal is an analog video output from the internal color palette DAC connected to a 37.5-W resistor to ground. This is used to provide noise immunity. Refer to the *Intel® 835 Chipset Design Guide*. REDW can be left as NC ("Not Connected") if not implemented.
GREENW	O Analog	**GreenW (Analog Output):** This signal is an analog video output from the internal color palette DAC connected to a 37.5-Ω resistor to ground This is used to provide noise immunity. Refer to the *Intel® 835 Chipset Design Guide*. GREENW can be left as NC ("Not Connected") if not implemented.
BLUEW	O Analog	**BlueW (Analog Output)**: This signal is an analog video output from the internal color palette DAC connected to a 37.5 ohm resistor to ground. This is used to provide noise immunity. Refer to the *Intel® 835 Chipset Design Guide*. BLUEW can be left as NC ("Not Connected") if not implemented.

NOTE: Total pins for display section: 9

3.2.3 Display Control Signals

Table 11. Display Control Signal Descriptions

Pin Name	Type	Description
DDC1 CLK	I/O LVTTL	**DDC1 CLK:** The specific function is DDC1 CLK for CRT/analog display. This signal is tri-stated during a hard reset. DDC1 CLK needs to be pulled up if the signal is NOT used when using internal graphics device.
DDC1 DATA	I/O LVTTL	**DDC1 DATA:** The specific function is DDC1 DATA for CRT/analog display. This signal is tri-stated during a hard reset. DDC1 DATA needs to be pulled up if the signal is NOT used when using internal graphics device.
I2C CLK	I/O LVTTL	**I2C CLK**: The specific function is I2C CLK. This signal can be used as GMBUS bus for RGBA device. This signal is tri-stated during a hard reset. I2C CLK needs to be pulled up if the signal is NOT used when using internal graphics device.
I2C DATA	I/O LVTTL	**I2C DATA**: The specific function is I2C DATA. This signal can be used as GMBUS bus for RGBA device. This signal is tri-stated during a hard reset. I2C DATA needs to pull up if the signal is NOT used when using internal graphics device.
DDC2 CLK	I/O LVTTL	**DDC2 CLK**: The specific function is DDC2 CLK for digital display, EDID info or as GMBUS bus for RGBA device. This signal is tri-stated during a hard reset. DDC2_CLK needs to be pull up if the signal is NOT used when using internal graphics device.
DDC2 DATA	I/O LVTTL	**DDC2 DATA**: The specific function is DDC2 DATA for digital display, EDID information or as GMBUS bus for RGBA device. This signal is tri-stated during a hard reset. DDC2_DATA needs to be pull up if the signal is NOT used when using internal graphics device.

NOTE: Total pins for this section: 6

3.3 Voltage References, PLL Power

Table 12. Voltage References, PLL Power Signal Descriptions (Sheet 1 of 2)

Signal Name	Number	Description
H_GTL REF [1:0]	2	**GTL Reference**: Reference voltage input for the host AGTL interface. GTLREF is 2/3 * VTT. VTT is nominally 1.25 V.
VTT	9	**Host Voltage**: VTT is nominally 1.25 V for host signals.
RGBAREF	1	**RGBA Reference**: Reference voltage input for the RGBA interface. RGBAREF is 0.5 * Vdd when Vdd=1.5 V.
VCCP 15G	8	**RGBA Voltage**: VDD is nominally 1.5 V.
VCCQ 15G	2	**RGBA Quiet Voltage**: Quiet voltage is also 1.5 V.
HLREF	1	**Hub Interface Reference**: Reference voltage input for the hub interface. HLREF is 0.5 * Vdd.

Table 12. Voltage References, PLL Power Signal Descriptions (Sheet 2 of 2)

Signal Name	Number	Description
VCCP HUB	2	**Hub Interface Voltage**: VCC supplies for the hub interface are 1.8 V.
SM VREF [1:0]	2	**System Memory Reference**: Reference voltage input for system memory is VCCP SM/6 = 0.55 V.
VCCP SM	14	**System Memory Voltage**: VCC supplies for system memory are 3.3 V.
VCCQ SM	5	**System Memory Quiet Voltage**: Quiet VCC for the system memory interface is 3.3 V.
VCCP GPIO	2	**GPIO Voltage**: VCC supplies for general purpose I/O signals are 3.3 V.
VCCP DVO	3	**RGBA Voltage**: VCC supplies for digital video output signals are 1.5 V.
VCCA DAC; VSSA DAC	3	**DAC Voltage**: VCCA and VSSA supplies for the DAC. VCCA DAC is 1.8 V.
RAM REF	2	**Reserved (Rambus* Reference)**: Reference voltage input for the Rambus RSL interface. RAMREF is approximately 1.4 V. Rambus is no longer supported.
VCCP CMOS	3	**Reserved (Graphics Memory CMOS Voltage)**: VCC and VSS supplies for local memory CMOS signals. VCCP CMOS is 1.8 V. Local memory is no longer supported.
VCCLM	9	**Reserved (VCC Graphics Memory Voltage)**: VCC supplies for local memory. VCC_LM is 1.8 V. Local memory is no longer supported.
VCCR	5	**Reserved (VDD Graphics Memory Voltage)**: VDD supplies for local memory. VCCR is 1.25 V. Local memory no longer supported.
VDD	2	**Reserved (VDD Graphics Memory Voltage)**: VDD supplies for local memory. VDD is 1.25 V. Local memory no longer supported.
VCC 18	1	**Core Voltage**: VCC 18 is 1.8 V.
VCCA CPLL; VSSA CPLL	2	**Graphics Core PLL Voltage**: VCCA and VSSA supplies for core PLL. VCCA CPLL is 1.25 V.
VCCA HPLL; VSSA HPLL	2	**Host/Memory/Hub/RGBA PLL Voltage**: VCCA and VSSA supplies for host PLL. VCCA HPLL is 1.25 V.
VCCA DPLL [1:0]; VSSA DPLL [1:0]	4	**Display PLL Voltage**: VCCA and VSSA supplies for display PLL. VCCA DPLL is 1.25 V.
VCC	24	**Core VCC**: 1.25 V.
VSS	140	**Ground pins**.

NOTE: Total pins for this section: 248

4.0 Register Description

This section details register access and provides PCI register address maps.

4.1 Conceptual Overview of the Platform Configuration Structure

The Intel® 835 chipset GMCH and the ICH4 are physically connected with the hub interface. From a configuration standpoint the hub interface connecting the GMCH and the ICH4 is logically PCI bus #0. All devices internal to the GMCH and ICH4 appear to be on PCI bus #0. The system's primary PCI expansion bus is physically attached to the ICH4, and from a configuration standpoint appears as a hierarchical PCI bus behind a PCI-to-PCI bridge. The primary PCI expansion bus connected to the ICH4 has a programmable PCI bus number.

Note: Even though the primary PCI bus is referred to as PCI0 in this document it is not PCI bus #0 from a configuration standpoint.

The GMCH contains three PCI devices within a single physical component. The configuration registers for Device 0 and 1 are mapped as devices residing on PCI bus #0.

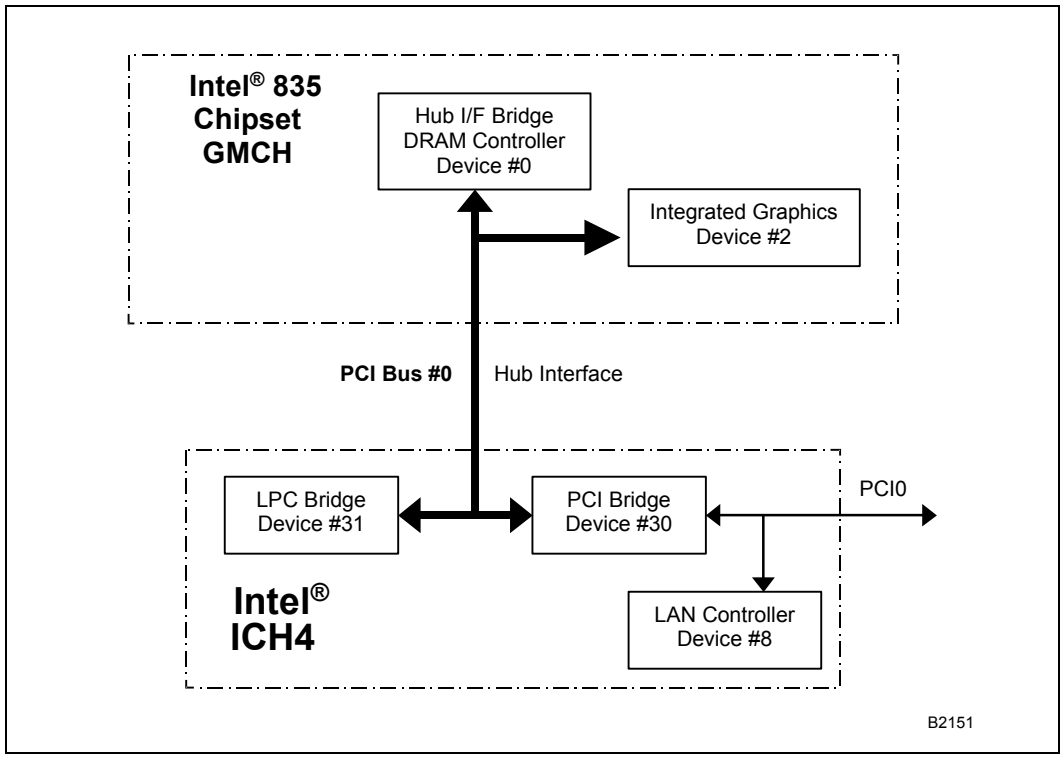
- **Device 0: Host-Hub Interface Bridge/SDRAM Controller**. Logically, this appears as a PCI device residing on PCI bus #0. Physically, device 0 contains the standard PCI registers, SDRAM registers, the graphics aperture controller, and other GMCH specific registers.
- **Device 1: Host-AGP Bridge**[1]. Logically, this appears as a "virtual" PCI-to-PCI bridge residing on PCI bus #0. Physically, device 1 contains the standard PCI-to-PCI bridge registers and the standard PCI1 configuration registers.
- **Device 2: Integrated Graphics Device (IGD).** Logically, this appears as a PCI device residing on PCI bus #0. Physically, device 2 contains the standard registers of a PCI graphics controller device.

Logically, the ICH4 appears as two PCI devices within a single physical component also residing on PCI bus #0. One of the ICH4 devices residing on PCI bus #0 is a PCI-to-PCI bridge. Logically, the primary side of the bridge resides on PCI bus #0 while the secondary side is the standard PCI expansion bus (PCI0). Also within the ICH4 is another PCI device, the LAN controller, which resides on the standard PCI expansion bus (PCI0) down from the PCI-to-PCI bridge.

Note: A physical PCI bus #0 does not exist and the hub interface and the internal devices in the GMCH and ICH4 logically constitute PCI bus #0 to configuration software. This is shown in Figure 2.

1. AGP capabilities are not functional in the Intel® 835 chipset.

Figure 2. Logical Bus Structure During PCI Configuration

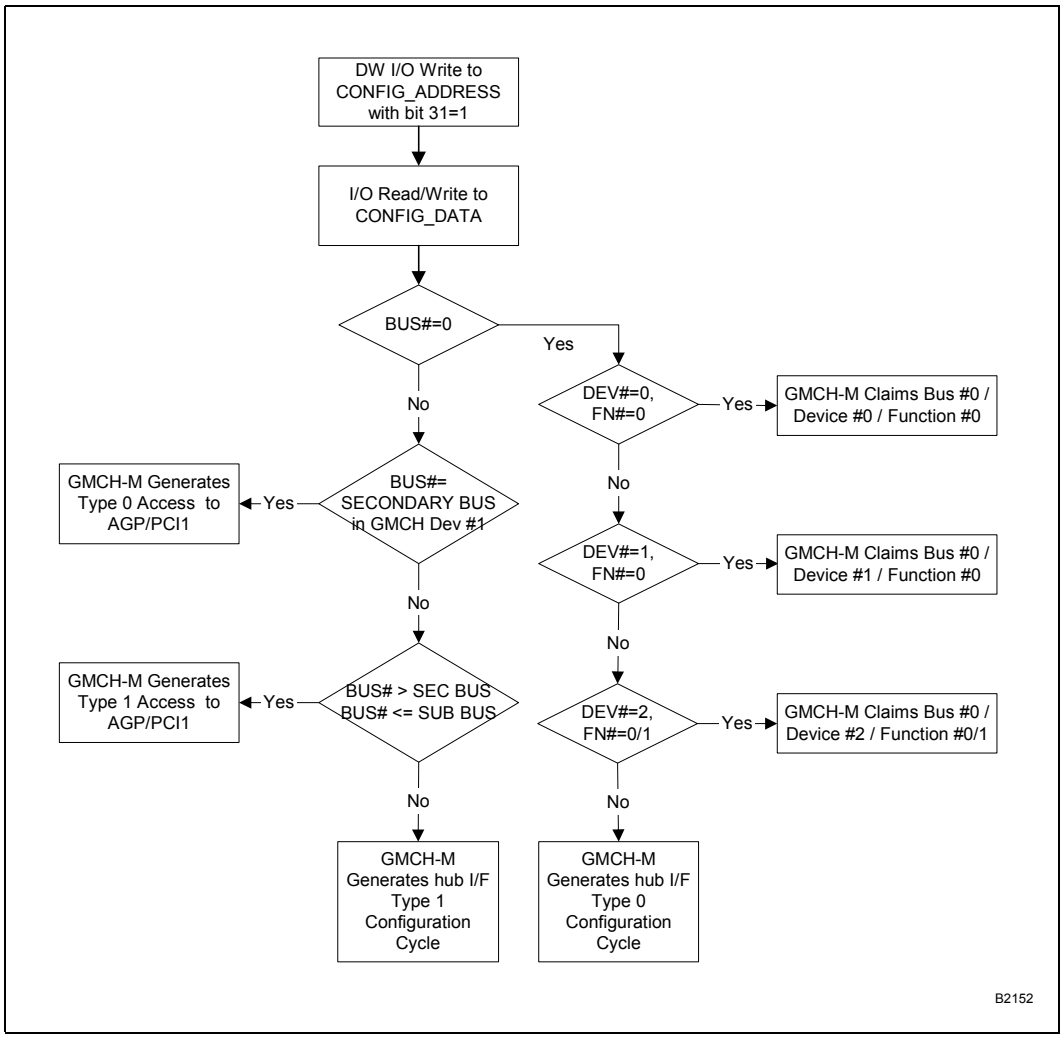


4.2 Routing Configuration Accesses to PCI0

The Intel 835 chipset GMCH supports two bus interfaces: hub interface and PCI. PCI configuration cycles are selectively routed to both interfaces. The GMCH is responsible for routing PCI configuration cycles to the proper interface. PCI configuration cycles to ICH4 internal devices and primary PCI devices (including downstream devices) are routed to the ICH4 via the hub interface.

4.2.1 Intel® 835 Chipset GMCH Configuration Cycle Flow Chart

Figure 3. Configuration Cycle Flow Chart



A detailed description of the mechanism for translating CPU I/O bus cycles to configuration cycles on one of the two buses is described in Figure 3.

4.2.2 PCI Bus Configuration Mechanism

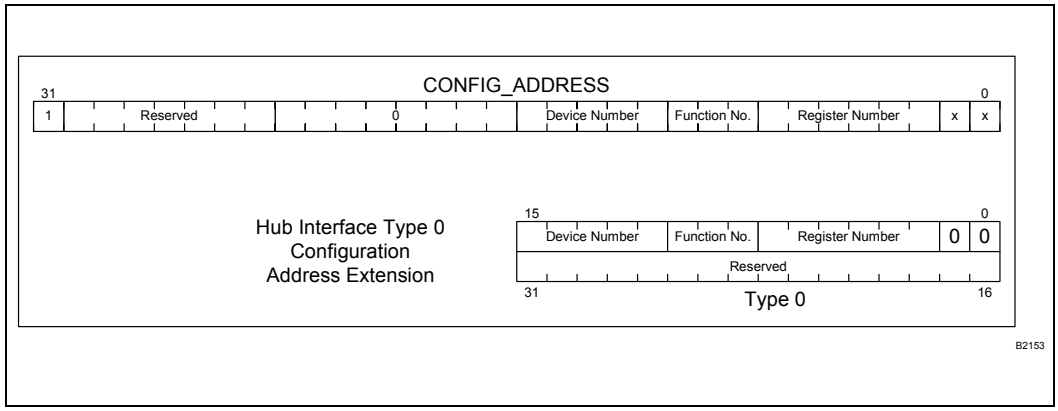
The PCI bus defines a slot based “configuration space” that allows each device to contain up to eight functions with each function containing up to 256, 8-bit configuration registers. The PCI specification defines two bus cycles to access the PCI configuration space: configuration read and configuration write. Memory and I/O spaces are supported directly by the CPU. Configuration space is supported by a mapping mechanism implemented within the GMCH. The PCI specification defines two mechanisms to access configuration space, mechanism #1 and mechanism #2. The GMCH supports only mechanism #1 for PCI configuration accesses.

The configuration access mechanism makes use of the CONFIG_ADDRESS register and CONFIG_DATA register. To reference a configuration register, a DWORD I/O write cycle is used to place a value into CONFIG_ADDRESS that specifies the PCI bus, the device on that bus, the function within the device, and a specific configuration register of the device function being accessed. First CONFIG_ADDRESS[31] must be valid to enable a configuration cycle. Second, CONFIG_DATA becomes a window into the four bytes of configuration space specified by the contents of CONFIG_ADDRESS. Any read or write to CONFIG_DATA results in the Intel 835 chipset GMCH translating the CONFIG_ADDRESS into the appropriate configuration cycle. The GMCH is responsible for translating and routing the CPU's I/O accesses to the CONFIG_ADDRESS and CONFIG_DATA registers to internal GMCH configuration registers or hub interface.

4.2.3 PCI Bus #0 Configuration Mechanism

The Intel 835 chipset GMCH decodes the bus number (bits 23:16) and the device number fields of the CONFIG_ADDRESS register. When the bus number field of CONFIG_ADDRESS is zero the configuration cycle is targeting a PCI bus #0 device. The host-hub interface bridge entity within the GMCH is hardwired as device #0 on PCI bus #0. The host-PCI1 bridge entity within the GMCH is hardwired as device #1 on PCI bus #0 (not functional in the Intel 835 chipset). The integrated graphics entity within the GMCH is hardwired as device #2 on PCI bus #0. Configuration cycles to the GMCH internal devices are confined to the GMCH and not sent over hub interface. Accesses to devices #3 to #31 are forwarded over hub interface as type 0 configuration cycles (see hub interface specifications). A [1:0] of the hub interface request packet for the type 0 configuration cycle is "00." Bits 15:2 of the CONFIG_ADDRESS register are translated to the A [15:2] field of the hub interface request packet of the configuration cycle as shown in Figure 4. The ICH4 decodes the type 0 access and generates a configuration access to the selected internal device.

Figure 4. Hub Interface Type 0 Configuration Address Translation

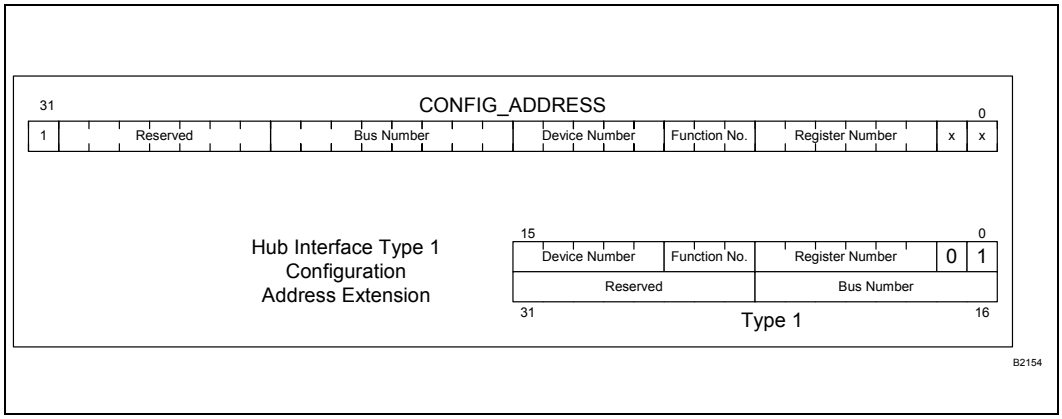


4.2.4 Primary PCI and Downstream Configuration Mechanism

If the bus number in the CONFIG_ADDRESS is non-zero, and is less than the value programmed into the Intel 835 chipset GMCH's device #1 SECONDARY BUS NUMBER register or greater than the value programmed into the SUBORDINATE BUS NUMBER register, the GMCH generates a type 1 hub interface configuration cycle. A [1:0] of the hub interface request packet for the type 1 configuration cycle is "01." Bits 31:2 of the CONFIG_ADDRESS register are translated to the A [31:2] field of the hub interface request packet of the configuration cycle as shown in

Figure 5. The ICH4 compares the non-zero bus number with the SECONDARY BUS NUMBER and SUBORDINATE BUS NUMBER registers of its P2P bridges to determine if the configuration cycle is meant for primary PCI, one of the ICH4's hub interfaces, or a downstream PCI bus.

Figure 5. Hub Interface Type 1 Configuration Address Translation



4.2.5 Internal GMCH Configuration Register Access Mechanism

Accesses decoded as PCI bus #0/device #0 (host-hub interface bridge/SDRAM controller), PCI bus #0/device #1 (host-PCI1 bridge), or PCI bus #0/device#2 (integrated graphics device) are sequenced as type 0 PCI configuration cycle accesses on bus #0 to device #0/function #0, device #1/function #0, and device #2/function #0/1. Note that because GMCH device #0 and #1 are not multi-function devices, the function number should always be '0.' If the function number is not '0' for accesses to device #0 or #1, the GMCH does not claim the configuration cycle and it is forwarded to the hub interface where it should be master aborted (by the ICH4) in the same way as transactions to other unimplemented PCI configuration targets.

4.3 GMCH Register Introduction

The Intel 835 chipset GMCH contains two sets of software-accessible registers, accessed via the host CPU I/O address space:

- Control registers I/O mapped into the CPU I/O space, which control access to PCI configuration space (see Section 4.4).
- Internal configuration registers residing within the GMCH that are partitioned into two logical device register sets ("logical" because they reside within a single physical device). The first register set is dedicated to host-hub interface bridge functionality (controls PCI bus #0, i.e., SDRAM configuration, other chipset operating parameters and optional features). The second register block is dedicated to the Integrated Graphics Device (IGD) function.

This configuration scheme is necessary to accommodate the existing and future software configuration model supported by Microsoft* where the host bridge functionality is supported and controlled through a dedicated specific driver. Virtual PCI-PCI bridge functionality is supported through standard PCI bus enumeration configuration software. The term "virtual" is used to indicate that no real physical embodiment of the PCI-PCI bridge functionality exists within the GMCH, but that the GMCH's internal configuration register sets are organized in this manner to create that impression to the standard configuration software.

The GMCH supports PCI configuration space accesses using the mechanism denoted as configuration mechanism #1 in the PCI specification. The GMCH internal registers (both I/O mapped and configuration registers) are accessible by the host CPU. The registers can be accessed as byte, word (16-bit), or DWORD (32-bit) quantities, with the exception of CONFIG_ADDRESS that can only be accessed as a DWORD. All multi-byte numeric fields use "little-endian" ordering (i.e., lower addresses contain the least significant parts of the field).

Some of the GMCH registers described in this section contain reserved bits. These bits are labeled "Reserved." Software must deal correctly with fields that are reserved. On read operations, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On write operations, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. Note that the software does not need to perform read, merge, and write operations for the configuration address register.

In addition to reserved bits within a register, the GMCH contains address locations in the configuration space of the host-hub interface bridge entity that are marked either "Reserved" or "Intel Reserved." The GMCH responds to accesses to "Reserved" address locations by completing the host cycle. When a "Reserved" register location is read, a zero value is returned. Writes to "Reserved" registers have no effect on the GMCH. Registers that are marked as "Intel Reserved" must not be modified by system software. Writes to "Intel Reserved" registers may cause system failure. Reads to "Intel Reserved" registers may return a non-zero value. "Reserved" registers can be 8-, 16-, or 32-bit in size.

Upon reset, the GMCH sets all of its internal configuration registers to predetermined default states. Some register values at reset are determined by external strapping options. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. It is the responsibility of the system initialization software (usually BIOS) to properly determine the SDRAM configurations, operating parameters and optional system features that are applicable, and to program the GMCH registers accordingly.

4.4 I/O Mapped Registers

The Intel 835 chipset GMCH contains a set of registers that reside in the CPU I/O address space – the configuration address (CONFIG_ADDRESS) register and the configuration data (CONFIG_DATA) register. The configuration address register enables/disables the configuration space and determines what portion of configuration space is visible through the configuration data window.

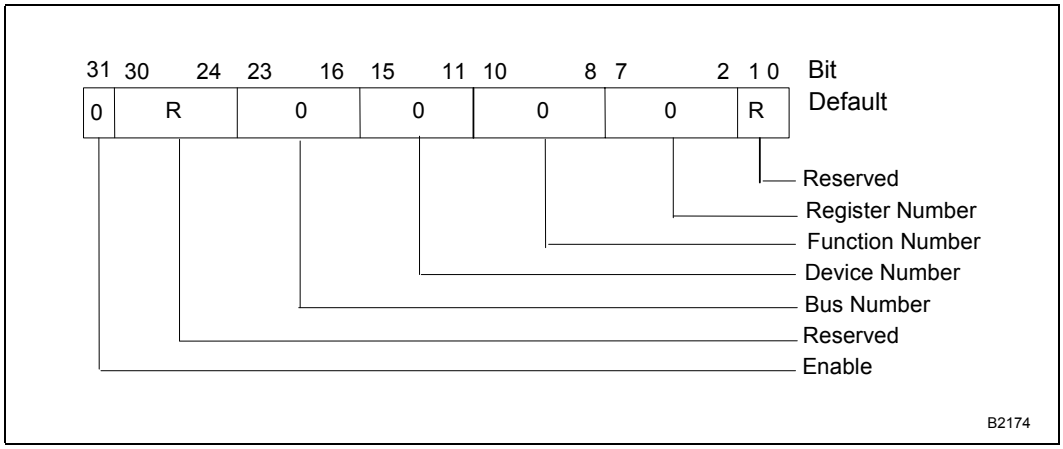
4.4.1 CONFIG_ADDRESS - Configuration Address Register

Table 13. CONFIG_ADDRESS - Configuration Address Register

I/O Address:	0CF8h Accessed as a DWORD	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32-bit

CONFIG_ADDRESS is a 32-bit register accessed only when referenced as a DWORD. A byte or word reference "passes through" the configuration address register and hub interface onto the PCI0 bus as an I/O cycle. The CONFIG_ADDRESS register contains the bus number, device number, function number, and register number for which a subsequent configuration access is intended.

Figure 6. CONFIG_ADDRESS Register



CONFIG_ADDRESS - Configuration Address Register (Sheet 1 of 2)	
Bits	**Description**
31	**Configuration Enable (CFGE)**. When this bit is set to one, accesses to PCI configuration space are enabled. If this bit is reset to zero, accesses to PCI configuration space are disabled.
30:24	**Reserved** (These bits are read-only and have a value of 0).
23:16	**Bus Number**. When the bus number is programmed to 00h the target of the configuration cycle is either the GMCH or the ICH4. The configuration cycle is forwarded to hub interface if the bus number is programmed to 00h and no device internal to the GMCH is the target. If the bus number is non-zero and matches the value programmed into the SECONDARY BUS NUMBER register of the PCI1 bridge, a type 0 PCI configuration cycle is generated on PCI1. If the bus number is non-zero, greater than the value in the SECONDARY BUS NUMBER register of the PCI1 bridge, and less than or equal to the value programmed into the SUBORDINATE BUS NUMBER register, a type 1 PCI configuration cycle is generated on PCI1. If the bus number is non-zero, and is less than the value programmed into the SECONDARY BUS NUMBER register of the PCI1 bridge, or is greater than the value programmed into the SUBORDINATE BUS NUMBER register, a type 1 hub interface configuration cycle is generated.
15:11	**Device Number**. This field selects one agent on the PCI bus selected by the bus number. When the bus number field is "00" the GMCH decodes the device number field. The GMCH is always device #0 for the host-hub interface bridge entity, device #1 for the host-PCI1 entity, and device #2 for the integrated graphics device entity. Therefore, when the bus number = 0 and the device number = 0, 1, or 2, the internal GMCH devices are selected. If the bus number is non-zero and matches the value programmed into the SECONDARY BUS NUMBER register of the PCI1 bridge, a type 0 PCI configuration cycle is generated on PCI1. The device number field is decoded and the GMCH asserts one and only one GADxx signal as an IDSEL. GAD11 is asserted to access device #0, GAD12 for device #1, and so forth up to device #20 for which asserts GAD31. All device numbers higher than 20 cause a type 0 configuration access with no IDSEL asserted, which results in a master abort reported in the GMCH's "virtual" PCI-PCI bridge registers. For bus numbers resulting in PCI1 type 1 configuration cycles, the device number is propagated as GAD[15:11].

CONFIG_ADDRESS - Configuration Address Register (Sheet 2 of 2)	
Bits	**Description**
10:8	**Function Number**. This field is mapped to GAD[10:8] during PCI1 Configuration cycles. This allows the configuration registers of a particular function in a multi-function device to be accessed. The GMCH ignores configuration cycles to devices 1 if the function number is not equal to 0.
7:2	**Register Number**. This field selects one register within a particular bus, device, and function as specified by the other fields in the configuration address register. This field is mapped to GAD[7:2] during PCI1 Configuration cycles.
1:0	**Reserved**

4.4.2 CONFIG_DATA - Configuration Data Register

CONFIG_DATA is a 32-bit read/write window into configuration space. The portion of configuration space that is referenced by CONFIG_DATA is determined by the contents of CONFIG_ADDRESS.

Figure 7. CONFIG_DATA Register

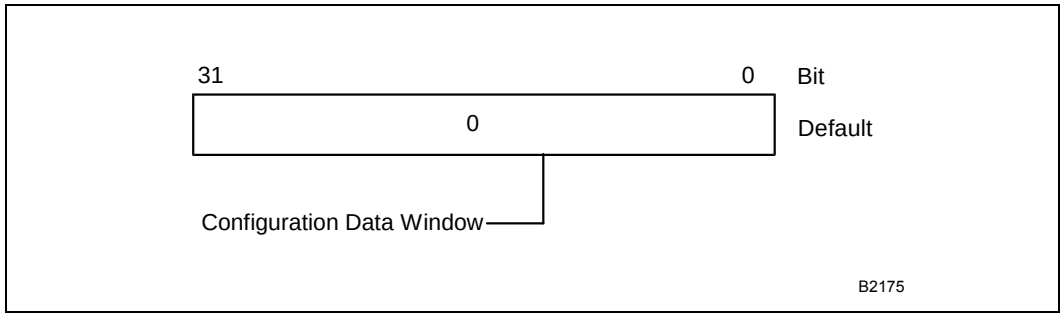


Table 14. CONFIG_DATA - Configuration Data Register

I/O Address:	0CFCh	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32-bit

Bit	Descriptions
31:0	**Configuration Data Window (CDW)**. If bit 31 of CONFIG_ADDRESS is 1, any I/O access to the CONFIG_DATA register is mapped to configuration space using the contents of CONFIG_ADDRESS.

4.5 GMCH Internal Device Registers

Table 15 shows the nomenclature of access attributes for the configuration space of each device.

Table 15. Nomenclature for Access Attributes

RO	**Read-Only**. If a register is read-only, writes to this register have no effect.
R/W	**Read/Write**. A register with this attribute can be read and written.
R/WC	**Read/Write Clear**. A register bit with this attribute can be read and written. However, a write of a 1 clears (sets to 0) the corresponding bit and a write of a 0 has no effect.
R/WO	**Read/Write Once**. A register bit with this attribute can be written to only once after power up. After the first write, the bit becomes read-only.
L	**Lock**. A register bit with this attribute becomes Read-Only after a lock bit is set.

4.5.1 SDRAM Controller/Host-hub Interface Device Registers - Dev #0

Table 16 shows the GMCH configuration space for device #0. An "s" in the default value field means that a strap determines the power-up default value for that bit.

Table 16. Host-Hub I/F Bridge/SDRAM Controller Configuration Space (Device #0) (Sheet 1 of 2)

Address Offset	Register Symbol	Register Name	Default Value	Access
00-01h	VID	Vendor Identification	8086h	RO
02-03h	DID	Device Identification	3579h	RO
04-05h	PCICMD	PCI Command	0006h	R/W
06-07h	PCISTS	PCI Status	0010h	RO, R/WC
08h	RID	Revision Identification	00h	RO
09h	-	Intel Reserved	-	-
0Ah	SUBC	Sub-Class Code	00h	RO
0Bh	BCC	Base Class Code	06h	RO
0Ch	-	Intel Reserved	-	-
0Dh	MLT	Master Latency Timer	00h	RO
0Eh	HDR	Header Type	00h	RO
0Fh	-	Intel Reserved	-	-
10-13h	APBASE	Aperture Base Configuration	00000008h	R/W, RO
14-2Bh	-	Intel Reserved	-	-
2C-2Dh	SVID	Subsystem Vendor Identification	0000h	R/WO
2E-2Fh	SID	Subsystem Identification	0000h	R/WO
30-33h	-	Intel Reserved	-	-
34h	CAPPTR	Capabilities Pointer	40h	RO
35-3Fh	-	Intel Reserved	-	-
40-44h	-	Intel Reserved	-	-
45-47h	-	Intel Reserved	-	-
48-4Bh	RRBAR	Register Range Base Address	00000000h	R/W, RO
4C-4Fh	-	Intel Reserved	-	-

Table 16. Host-Hub I/F Bridge/SDRAM Controller Configuration Space (Device #0) (Sheet 2 of 2)

Address Offset	Register Symbol	Register Name	Default Value	Access
50-51h	GCC0	GMCH Control Register 0	A072h	R/W, RO
52-53h	GCC1	GMCH Control Register 1	0000h	R/W
54-55h	-	Intel Reserved	-	-
56-57h	-	Intel Reserved	-	-
58h	FDHC	Fixed DRAM Hole Control	00h	R/W
59-5Fh	PAM [6:0]	Programmable Attribute Map (7 registers)	00h	R/W
60-67h	DRB [7:0]	DRAM Row Boundary (8 registers)	00h	R/W/L
68-6Fh	-	Intel Reserved	-	-
70-71h	DRA (1:0)	DRAM Row Attributes (2 registers)	FFh	R/W/L
72-77h	-	Intel Reserved	-	-
78-7Bh	DRT	DRAM Timing	00000010h	R/W
7C-7Fh	DRC	DRAM Controller Mode	00000000h	R/W
80-8Bh	-	Intel Reserved	-	-
8C-8Fh	DTC	DRAM Throttling Control	00000000h	R/W/L
90h	SMRAM	System Management RAM Control	02h	R/W/L
91h	ESMRAMC	Extended System Management RAM Control	38h	R/W
92-93h	ERRSTS	Error Status	0000h	R/WC
94-95h	ERRCMD	Error Command	0000h	R/W
96h	-	Intel Reserved	-	-
97h	-	Intel Reserved	-	-
98-9F	-	Intel Reserved	-	-
A0-A3h	ACAPID	Reserved	00200002h	RO1
A4-A7h	AGPSTAT	Reserved	1F000217h	RO1
A8-ABh	AGPCMD	Reserved	00000000h	RW1
AC-AFh	-	Intel Reserved	-	-
B0-B1h	AGPCTRL	Reserved	0000h	R/W1
B2-B3h	AFT	Reserved	0000h	R/W1
B4h	APSIZE	Reserved	00h	R/W1
B5-B7h	-	Intel Reserved	-	-
B8-BBh	ATTBASE	Reserved	00000000h	R/W1
BCh	AMTT	Reserved	00h	R/W1
BDh	LPTT	Reserved	00h	R/W1
BE-BFh	-	Intel Reserved	-	-
C2-EBh	-	Intel Reserved	-	-
EC-EFh	BUFF_SC	System Memory Buffer Strength Control	00000000h	R/W
F0-FFh	-	Intel Reserved	-	-

NOTES:
1. Register has no functionality in the Intel 835 chipset.
2. Key:
   RO = Read-Only
   WO = Write-Only
   WC = Write clear
   W = Write
   L = Read-Only if D_LCK=1

4.5.1.1 VID - Vendor Identification Register - Dev #0

The VID register contains the vendor identification number. This 16-bit register combined with the device identification register uniquely identifies any PCI device. Writes to this register have no effect.

Table 17. VID - Vendor Identification Register - Dev #0

Offset:	00 - 01h	***Attribute:***	Read-Only
Default Value:	8086h	***Size:***	16-bit

Bits	Description
15:0	**Vendor Identification Number**. This is a 16-bit value assigned to Intel. Intel VID = 8086h. Default value=1000/0000/1000/0110.

4.5.1.2 DID - Device Identification Register - Dev #0

This 16-bit register combined with the vendor identification register uniquely identifies any PCI device. Writes to this register have no effect.

Table 18. DID - Device Identification Register - Dev #0

Offset:	02 - 03h	***Attribute:***	Read-Only
Default Value:	3579h	***Size:***	16-bit

Bits	Description
15:0	**Device Identification Number**. This is a 16-bit value assigned to the GMCH host-hub interface bridge, device #0. Default value=0011/0101/0111/1001.

4.5.1.3 PCICMD - PCI Command Register - Dev #0

Because GMCH device #0 is the host-to-hub interface bridge, many of the PCI-specific bits in this register do not apply.

Table 19. PCICMD - PCI Command Register - Dev #0

	Offset: 04-05h	***Attribute:***	Read/Write
	Default Value: 0006h	***Size:***	16-bit

Bits	Description
15:10	**Reserved**
9	**Fast Back-to-Back**. This bit controls whether or not the master can do fast back-to-back write. Because device #0 is strictly a target this bit is not implemented and is hardwired to zero. Writes to this bit position have no effect. Default value=0.
8	**SERR Enable (SERRE)**. This bit is a global enable bit for device #0 SERR messaging. The GMCH does not have a SERR# signal. The GMCH communicates the SERR# condition by sending a SERR message to the ICH4. If this bit is set to one, the GMCH is enabled to generate SERR messages over hub interface for specific device #0 error conditions that are individually enabled in the ERRCMD register. The error status is reported in the ERRSTS and PCISTS registers. If SERRE is reset to zero, then the SERR message is not generated by the GMCH for device #0. **NOTE**: This bit only controls SERR messaging for the device #0. device #1 has its own SERRE bit to control error reporting for error conditions occurring on device #1. The two control bits are used in a logical OR manner to enable the SERR hub interface message mechanism. Default value=0.
7	**Address/Data Stepping**. Address/data stepping is not implemented in the GMCH, and this bit is hardwired to zero. Writes to this bit position have no effect. Default value=0.
6	**Parity Error Enable (PERRE)**. PERR# is not implemented by the GMCH, and this bit is hardwired to zero. Writes to this bit position have no effect. Default value=0.
5	**VGA Palette Snoop**. The GMCH does not implement this bit and it is hardwired to zero. Writes to this bit position have no effect. Default value=0.
4	**Memory Write and Invalidate Enable**. The GMCH never uses this command and this bit is hardwired to zero. Writes to this bit position have no effect. Default value=0.
3	**Special Cycle Enable**. The GMCH does not implement this bit and it is hardwired to zero. Writes to this bit position have no effect. Default value=0.
2	**Bus Master Enable (BME)**. The GMCH is always enabled as a master on hub interface. This bit is hardwired to one. Writes to this bit position have no effect. Default value=1.
1	**Memory Access Enable (MAE)**. The GMCH always allows access to main memory. This bit is not implemented and is hardwired to one. Writes to this bit position have no effect. Default value=1.
0	**I/O Access Enable (IOAE).** This bit is not implemented in the GMCH and is hardwired to zero. Writes to this bit position have no effect. Default value=0.

4.5.1.4 PCISTS - PCI Status Register - Dev #0

PCISTS is a 16-bit status register that reports the occurrence of error events on device #0's hub interface. Bit 14 is read/write clear. All other bits are read-only. Because GMCH device #0 is the host-to-hub interface bridge, many of the PCI specific bits in this register do not apply.

Table 20. PCISTS - PCI Status Register - Dev #0

	Offset: 06-07h	***Attribute:***	Read-Only, Read/Write Clear
	Default Value: 0010h	***Size:***	16-bit

Bits	Description
15	**Detected Parity Error (DPE)**. This bit is hardwired to zero. Writes to this bit position have no effect. Default value=0.
14	**Signaled System Error (SSE).** This bit is set to one when GMCH device #0 generates a SERR message over the hub interface for any enabled device #0 error condition. Device #0 error conditions are enabled in the PCICMD and ERRCMD registers. Device #0 error flags are read/reset from the PCISTS or ERRSTS registers. Software sets SSE to zero by writing one to this bit. Default value=0.
13	**Received Master Abort Status (RMAS).** This bit is set when the GMCH generates a hub interface request that receives a master abort completion packet. Software clears this bit by writing one to it. Default value=0.
12	**Received Target Abort Status (RTAS**). This bit is set when the GMCH generates a hub interface request that receives a Target Abort completion packet. Software clears this bit by writing one to it. Default value=0.
11	**Signaled Target Abort Status (STAS)**. The GMCH does not generate a target abort hub interface completion packet. This bit is not implemented in the GMCH and is hardwired to zero. Writes to this bit position have no effect. Default value=0.
10:9	**DEVSEL# Timing (DEVT).** Hub interface does not comprehend DEVSEL# protocol. These bits are hardwired to “00.” Writes to these bits have no effect. Default value=00.
8	**Data Parity Detected (DPD)**. GMCH does not support parity on the hub interface. This bit is hardwired to zero. Writes to this bit position have no effect. Default value=0.
7	**Fast Back-to-Back (FB2B).** Hub interface does not comprehend PCI fast back-to-back protocol. This bit is hardwired to zero. Writes to this bit position have no effect. Default value=0.
6:5	**Reserved**
4	**Capability List (CLIST)**. This bit is hardwired to one to indicate to the configuration software that this device/function implements a list of new capabilities. A list of new capabilities is accessed via register CAPPTR at configuration address offset 34h. Register CAPPTR contains an offset pointing to the start address within configuration space of this device where the capabilities linked list begins. Default value=1.
3:0	**Reserved**

4.5.1.5 RID - Revision Identification Register - Dev #0

This register contains the revision number of the GMCH device #0. These bits are read-only and writes to this register have no effect.

Table 21. RID - Revision Identification Register - Dev #0

	Offset: 08h	***Attribute:***	Read-Only
	Default Value: 00h (A0 silicon)	***Size:***	8-bit

Bits	Description
7:0	**Revision Identification Number**. This is an 8-bit value that indicates the revision identification number for the GMCH device #0. See the specifications update for the latest silicon revision. Silicon Revision — Default Value A0 — 0000/0000 (00h)

4.5.1.6 SUBC - Sub-Class Code Register - Dev #0

This register contains the sub-class code for the GMCH device #0. This code is 00h indicating a host bridge device. The register is read-only.

Table 22. SUBC - Sub-Class Code Register - Dev #0

	Offset: 0Ah	***Attribute:***	Read-Only
	Default Value: 00h	***Size:***	8-bit

Bits	Description
7:0	**Sub-Class Code (SUBC)**. This is an 8-bit value that indicates the category of bridge into which the GMCH falls. The code is 00h indicating a host bridge. Default value=0000/0000.

4.5.1.7 BCC - Base Class Code Register - Dev #0

This register contains the base class code of the GMCH device #0. This code is 06h indicating a bridge device. This register is read-only.

Table 23. BCC - Base Class Code Register - Dev #0

	Offset: 0Bh	***Attribute:***	Read-Only
	Default Value: 06h	***Size:***	8-bit

Bits	Description
7:0	**Base Class Code (BASEC).** This is an 8-bit value that indicates the base class code for the GMCH. This code has the value 06h. Default value=0000/0110.

4.5.1.8 MLT - Master Latency Timer Register - Dev #0

Hub interface does not comprehend the concept of a master latency timer. Therefore the functionality of this register is not implemented and the register is hardwired to zero.

Table 24. MLT - Master Latency Timer Register - Dev #0

	Offset: 0Dh	***Attribute:***	Read-Only
	Default Value: 00h	***Size:***	8-bit

Bits	Description
7:0	These bits are hardwired to zero. Writes have no effect. Default value=0000/0000.

4.5.1.9 HDR - Header Type Register - Dev #0

This register identifies the header layout of the configuration space. No physical register exists at this location.

Table 25. HDR - Header Type Register - Dev #0

	Offset: 0Eh	***Attribute:***	Read-Only
	Default Value: 00h	***Size:***	8-bit

Bits	Description
7:0	This read-only field always returns zero when read; writes have no effect. Default value=0000/0000.

4.5.1.10 APBASE - Aperture Base Configuration Register - Dev #0

The APBASE is a standard PCI base address register that is used to set the base of the graphics aperture. The standard PCI configuration mechanism defines the base address configuration register such that only a fixed amount of space can be requested (dependent on which bits are hardwired to zero or behave as hardwired to zero).

To allow for flexibility of the aperture, an additional register called APSIZE is used as a “back-end” register to control which bits of the APBASE behave as hardwired to zero. This register is programmed by the GMCH-specific BIOS code that runs before any of the generic configuration software is run.

Note: Bit 9 of the GCC0 register at 51-50h is used to prevent accesses to the aperture range before the configuration software initializes this register and the appropriate translation table structure has been established in the main memory.

Table 26. APBASE - Aperture Base Configuration Register - Dev #0

Offset:	10-13h	*Attribute:*	Read/Write, Read-Only
Default Value:	00000008h	*Size:*	32-bit

Bits	Description
31:28	**Upper Programmable Base Address bits (R/W).** These bits are used to locate the range size selected via lower bits 27:4. Default Value = 0000.
27:25	**Lower "Hardwired"/Programmable Base Address bits**. These bits behave as a "hardwired" or as a programmable depending on the contents of the APSIZE register as defined below: **27 26 25 Aperture Size r/w** R/W R/W R/W 32 Mbytes R/W R/W R/W 64 Mbytes R/W 0 0 128 Mbytes 0 0 0 256 Mbytes The Default for APSIZE[5:3,0]=0000 with forces default APBASE[27:25] =000 (i.e., all bits respond as "hardwired" to zero). This provides a default to the maximum aperture size of 256MB. The GMCH specific BIOS is responsible for selecting smaller size (if required) before PCI configuration software runs and establishes the system address map. Default Value=000.
24:4	**Hardwired to Zero.** This forces minimum aperture size selected by this register to be 32 Mbytes.
3	**Prefetchable (RO).** This bit is hardwired to one to identify the graphics aperture range as a prefetchable, i.e., there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables, and the GMCH may merge processor writes into this range without causing errors.
2:1	**Type (RO).** These bits determine addressing type and they are hardwired to "00" to indicate that address range defined by the upper bits of this register can be located anywhere in the 32-bit address space. Default Value=00.

4.5.1.11 SVID - Subsystem Vendor ID - Dev #0

This value is used to identify the vendor of the subsystem.

Table 27. SVID - Subsystem Vendor ID - Dev #0

Offset:	2C-2Dh	*Attribute:*	Read/Write Once
Default Value:	0000h	*Size:*	16-bit

Bits	Description
15:0	**Subsystem Vendor ID (R/WO).** The default value is 00h. This field should be programmed during boot-up. After this field is written once, it becomes read-only. Default value=0000/0000/0000/0000.

4.5.1.12 SID - Subsystem ID - Dev #0

This value is used to identify a particular subsystem.

Table 28. SID - Subsystem ID - Dev #0

	Offset: 2E-2Fh	***Attribute:***	Read/Write Once
	Default Value: 0000h	***Size:***	16-bit

Bits	Description
15:0	**Subsystem ID (R/WO)**. The default value is 00h. This field should be programmed during boot-up. After this field is written once, it becomes read-only. Default value=0000/0000/0000/0000.

4.5.1.13 CAPPTR - Capabilities Pointer - Dev #0

The CAPPTR provides the offset that is the pointer to the location where the first capability register set is located.

Table 29. CAPPTR - Capabilities Pointer - Dev #0

	Offset: 34h	***Attribute:***	Read-Only
	Default Value: 40h	***Size:***	8-bit

Bits	Description
7:0	Pointer to the start of capabilities register block. The value in this field is 40h. Default value=0100/0000.

4.5.1.14 RRBAR - Register Range Base Address Register - Dev #0

This register requests a 256-Kbyte allocation for the device registers. The base address is defined by bits 31 to 18 and can be used to access device configuration registers. Only DWORD-aligned writes are allowed to this space. See the table below for address map within the 512-Kbyte space.

This addressing mechanism may be used to write to registers that modify the device address map (includes all the BARs, PAMs, SMM registers, and pre-allocated memory registers). However, before using or allowing the use of the modified address map the BIOS must synchronize using an I/O or read cycle.

Note: Bit 8 of the GCC0 register at 51-50h is used to prevent accesses to this range before the configuration software initializes this register.

Table 30. RRBAR - Register Range Base Address Register - Dev #0

Offset: 48-4Bh	***Attribute:*** Read/Write, Read-Only
Default Value: 00000000h	***Size:*** 32-bit

Bits	Description
31:18	**Memory Base Address-R/W**. Set by the OS, these bits correspond to address signals [31:18]. Default value=0000/0000/0000/0.
17:15	**Address Mask-RO**. Hardwired to zeros to indicate 512-Kbyte address range. The minimum size that can be requested by converting all these bits to R/W would be 64 Kbyte. Default value=000.
15:8	**Reserved.** Hardwired to 00h.
7:0	**Scratch Pad Size-RO,** Hardwired to “00h.” 00h = 256B FFh = 64 Kbyte Default value=0000/0000.

Address Range		Description
Sub Ranges		
00000h to 3FFFFh Device 0 Space	00000h to 0003Fh	Read-Only: Maps to 00-3Fh of device #0 P&P register space.
	00040h to 000FFh	Read/Write: Maps to 40-FFh of device #0 P&P register space.
	00100h to 3FEFFh	Read/Write: Extended register space. Reserved.
	3FF00h to 3FFFFh	Scratch pad registers: 256 B, D-word read/write-able.

4.5.1.15 GCC0 - GMCH Control Register #0 - Dev #0

Table 31. GCC0 - GMCH Control Register #0 - Dev #0 (Sheet 1 of 2)

Offset:	50-51h	***Attribute:***	Read/Write, Read-Only
Default Value:	A072h	***Size:***	16-bit

Bits	Description
15	**Reserved**
14:12	**Low Priority Grace Period**. This value is loaded in SDRAM arbiter when a request is ongoing and a higher priority request is presented to the arbiter. The arbiter continues to grant the first request for this specified number of page hits (1 Kbyte). If the first requester causes a page miss or stops requesting the arbiter switches to the higher priority requester. (A request equals an oct-word, also known as dual-oct byte). **000** = 00 **001** = 04 **010** = 08 (Default) **011** = 16 **100** = 24 **101** = 32 **110** = Reserved **111** = Reserved Default value=010. Recommended value with internal graphics = 010 →8
11	**Scratch Pad Enable**. When set to one, this bit allows the upper 256 bytes of device #0 RRBAR space to be mapped to scratch pad RAM in the device. After **D_LCK is set, this bit becomes read-only.** **NOTE**: The BIOS can use the scratch pad area when devices on the PCI1 bus are inactive. Default value=0.
10	**Reserved**
9	**Aperture Access Global Enable (R/W)**. This bit is used to prevent access to the aperture from any port (CPU, PCI0 or PCI1) before the aperture range is established by the configuration software and the appropriate translation table in the main SDRAM has been initialized. It must be set after the system is fully configured for aperture accesses. Default value=0.
8	**RRBAR Access Enable**. When set to one, this bit enables the RRBAR space. When set to zero, accesses do not decode to register range. Default value=0.
7	**Reserved**

Table 31. GCC0 - GMCH Control Register #0 - Dev #0 (Sheet 2 of 2)

	Offset: 50-51h	***Attribute:***	Read/Write, Read-Only
	Default Value: A072h	***Size:***	16-bit

Bits	Description
6:4	**IOQ Request Grant Ceiling**. This value is loaded in SDRAM arbiter when an IOQ request is granted. It provides a grant for the duration specified for as long as the request is active or until a fixed higher priority request needs to be serviced. **111** = Infinite Ceiling (Default) **110** = 64 **101** = 48 **100** = 32 **011** = 24 **010** = 16 **001** = 08 **000** = 04 Default value=111. Contact your Intel Field Representative for more information.
3:1	**Reserved**
0	**Reserved**. Must be zero.

4.5.1.16 GCC1- GMCH Control Register #1 - Dev #0

Table 32. GCC1- GMCH Control Register #1 - Dev #0 (Sheet 1 of 2)

	Offset: 52-53h	***Attribute:***	Read/Write
	Default Value: 0000h	***Size:***	16-bit

Bits	Description
15:7	**Reserved**
6:4	**Graphics Mode Select (GMS**). This field is used to select the amount of main memory that is pre-allocated to support the internal graphics device in VGA (non-linear) and native (linear) modes. These three bits are valid only when internal graphics is enabled. 000 = No memory pre-allocated **(graphics memory disabled) [Reserved]** 001 = No memory pre-allocated **(graphics memory enabled) [Reserved]** 010 = DVMT (UMA) mode, 512K of memory pre-allocated for frame buffer 011 = DVMT (UMA) mode, 1 Mbyte of memory pre-allocated for frame buffer 100 = DVMT (UMA) mode, 8 Mbytes of memory pre-allocated for frame buffer**)** Note: All other combinations are reserved. This register is locked and becomes read-only when the D_LCK bit in the SMRAM register is set. Default Value = 000
3	**Device #2 Disable** When set to one, this bit disables device #2 and all associated spaces. Default Value = 0

Table 32. GCC1- GMCH Control Register #1 - Dev #0 (Sheet 2 of 2)

	Offset: 52-53h	***Attribute:***	Read/Write
	Default Value: 0000h	***Size:***	16-bit

Bits	Description
2	**Device #2 Function 1 Enable** When set to one, enables the second function within device #2. Default Value = 0
1	**IGD VGA Disable (IVD)**. The Intel® 835 GMCH is not legacy VGA compatible. Write as one. Default Value = 0
0	**Device 2: Graphics Memory Aperture Size** (controls GMADR register in device# 2) 0 = 128 Mbytes 1 = 64 Mbytes Default Value = 0

These register bits control the theft of memory from main memory space for use as graphics memory. The memory for TSEG is pre-allocated first and then the graphics memory is pre-allocated. An example of this theft mechanism is:

TOM equals 64 Mbytes.

TSEG selected as 512 Kbytes.

Graphics memory selected as 1 Mbyte.

General System RAM available in system = 62.5 Mbyte

General System RAM Range00000000h to 03E7FFFFh

TSEG Address Range03F80000h to 03FFFFFFh

TSEG pre-allocated from03F80000h to 03FFFFFFh

Graphics memory pre-allocated from 03E80000h to 03F7FFFFh

VGA Memory and IO Space decode priority:

Integrated Graphics Device (IGD), device #2.

hub interface.

VGA Memory Space decode to IGD:

```
IF IGE = '1' AND IVD = '0' AND Device # 2 Mem_Access_En = '1'
AND MSRb1 = '1' AND →Additional Qualification Within IGD
Decode
```

Mem Access→ GR06(3:2)	A0000h - AFFFFh	B0000h - B7FFFh	B8000h-BFFFFh
“00”	IGD	IGD	IGD
“01”	IGD	P2P bridge or hub interface	P2P bridge or hub interface
“10”	P2P bridge or hub interface	IGD	P2P bridge or hub interface
“11”	P2P bridge or hub interface	P2P bridge or hub interface	IGD

```
ELSE defaults to hub interface.

    VGA IO space decode to IGD:

    IF IGE = '1' AND IVD = '0' AND Device # 2 IO_Access_En = '1' AND

    Additional Qualification within IGD decode.
```

IO Access →MSRb0	3CX	3DX	3B0-3BB	3BC-3BF
“0”	IGD	P2P bridge or hub interface	IGD	P2P bridge or hub interface
“1”	IGD	IGD	P2P bridge or hub interface	P2P bridge or hub interface

```
ELSE defaults to hub interface.
```

4.5.1.17 FDHC - Fixed DRAM Hole Control Register - Dev #0

This 8-bit register controls a single fixed SDRAM hole: 15-16 Mbytes.

Table 33. FDHC - Fixed DRAM Hole Control Register - Dev #0

Offset:	58h	***Attribute:***	Read/Write
Default Value:	00h	***Size:***	8-bit

Bits	Description
7	**Hole Enable (HEN).** This field enables a memory hole in SDRAM space. Host cycles matching an enabled hole are passed on to ICH4 through hub interface. Hub interface cycles matching an enabled hole are ignored by the GMCH. Note that a selected hole is not remapped. **Bit 7 Hole Enabled** 0 None 1 15-16 Mbytes (1 Mbyte) Default value=0.
6:0	**Reserved**

4.5.1.18 PAM(6:0) - Programmable Attribute Map Registers - Dev #0

Table 34. PAM(6:0) - Programmable Attribute Map Registers - Dev #0

Offset:	59 - 5Fh	***Attribute:***	Read/Write
Default Value:	00h	***Size:***	4 bits/register, 14 registers

The GMCH allows programmable memory attributes on 13 legacy memory segments of various sizes in the 640-Kbyte to 1-Mbyte address range. Seven Programmable Attribute Map (PAM) registers are used to support these features. Cacheability of these areas is controlled through the MTRR registers in the P6 processor. Two bits are used to specify memory attributes for each memory segment. These bits apply to host, PCI, and hub interface initiator accesses to the PAM areas.

These attributes are:

- **RE - Read Enable**. When RE = 1, the CPU read accesses to the corresponding memory segment are claimed by the GMCH and directed to main memory. Conversely, when RE = 0, the host read accesses are directed to PCI0.
- **WE - Write Enable.** When WE = 1, the host write accesses to the corresponding memory segment are claimed by the GMCH and directed to main memory. Conversely, when WE = 0, the host write accesses are directed to PCI0.

The RE and WE attributes permit a memory segment to be read-only, write-only, read/write, or disabled. For example, if a memory segment has RE = 1 and WE = 0, the segment is read-only.

Each PAM register controls two regions, typically 16 Kbytes in size. Each of these regions has a 4-bit field. The four bits that control each region have the same encoding and are defined in Table 35.

Table 35. Attribute Bit Assignment

Bits [7, 3] Reserved	Bits [6, 2] Reserved	Bits [5 1] WE	Bits [4, 0] RE	Description
X	X	0	0	**Disabled**. SDRAM is disabled and all accesses are directed to the hub interface. The GMCH does not respond as a PCI1 or hub interface target for any read or write access to this area.
X	X	0	1	**Read-Only**. Reads are forwarded to SDRAM and writes are forwarded to hub interface for termination. This write protects the corresponding memory segment. The GMCH responds as a PCI1 or hub interface target for read accesses but not for any write accesses.
X	X	1	0	**Write-Only**. Writes are forwarded to SDRAM and reads are forwarded to the hub interface for termination. The GMCH responds as a PCI1 or hub interface target for write accesses but not for any read accesses.
X	X	1	1	**Read/Write**. This is the normal operating mode of main memory. Both read and write cycles from the host are claimed by the GMCH and forwarded to SDRAM. The GMCH responds as a PCI1 or hub interface target for both read and write accesses.

As an example, consider a BIOS that is implemented on the expansion bus. During the initialization process, the BIOS can be shadowed in main memory to increase the system performance. When BIOS is shadowed in main memory, it should be copied to the same address location. To shadow the BIOS, the attributes for that address range should be set to write-only. The BIOS is shadowed by first doing a read of that address. This read is forwarded to the expansion bus. The host then writes the same address, which is directed to main memory. After the BIOS is shadowed, the attributes for that memory area are set to read-only so that all writes are forwarded to the expansion bus. Figure 8 and Table 36 show the PAM registers and the associated attribute bits.

Figure 8. PAM Registers

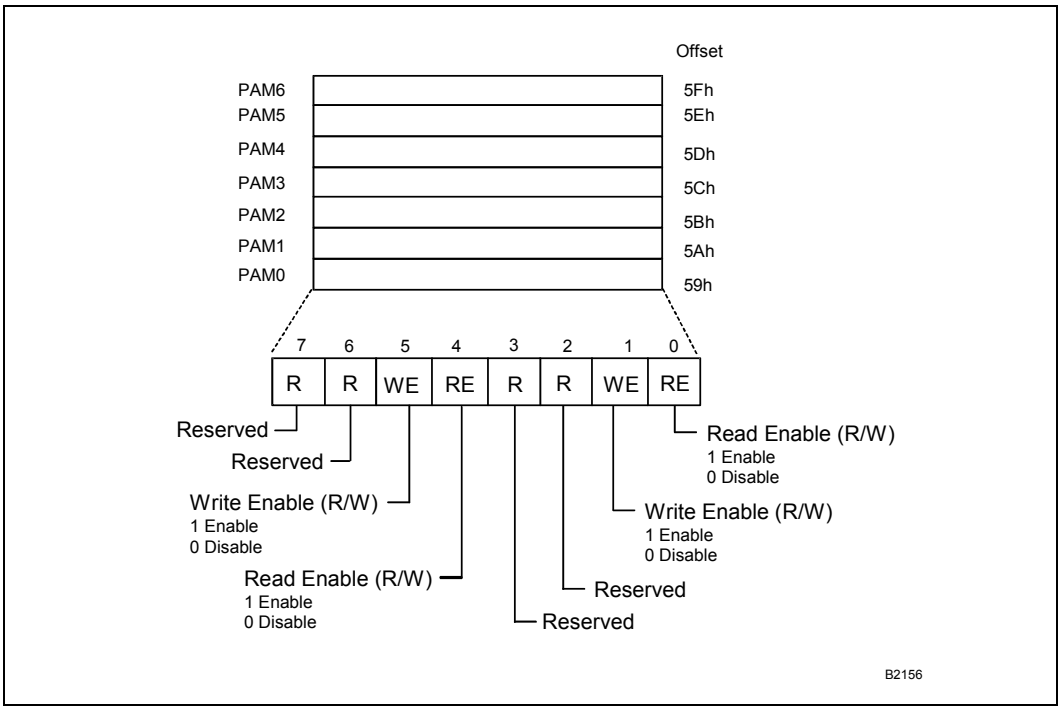


Table 36. PAM Registers and Associated Memory Segments (Sheet 1 of 2)

PAM Register	Attribute Bits	Memory Segment	Comments	Offset
PAM0[3:0]	Reserved			59h
PAM0[7:4]	R R WE RE	0F0000h - 0FFFFFh	BIOS Area	59h
PAM1[3:0]	R R WE RE	0C0000h - 0C3FFFh	ISA Add-on BIOS	5Ah
PAM1[7:4]	R R WE RE	0C4000h - 0C7FFFh	ISA Add-on BIOS	5Ah
PAM2[3:0]	R R WE RE	0C8000h - 0CBFFFh	ISA Add-on BIOS	5Bh
PAM2[7:4]	R R WE RE	0CC000h- 0CFFFFh	ISA Add-on BIOS	5Bh
PAM3[3:0]	R R WE RE	0D0000h- 0D3FFFh	ISA Add-on BIOS	5Ch
PAM3[7:4]	R R WE RE	0D4000h- 0D7FFFh	ISA Add-on BIOS	5Ch
PAM4[3:0]	R R WE RE	0D8000h- 0DBFFFh	ISA Add-on BIOS	5Dh
PAM4[7:4]	R R WE RE	0DC000h- 0DFFFFh	ISA Add-on BIOS	5Dh
PAM5[3:0]	R R WE RE	0E0000h- 0E3FFFh	BIOS Extension	5Eh

Table 36. PAM Registers and Associated Memory Segments (Sheet 2 of 2)

PAM Register	Attribute Bits	Memory Segment	Comments	Offset
PAM5[7:4]	R R WE RE	0E4000h- 0E7FFFh	BIOS Extension	5Eh
PAM6[3:0]	R R WE RE	0E8000h- 0EBFFFh	BIOS Extension	5Fh
PAM6[7:4]	R R WE RE	0EC000h- 0EFFFFh	BIOS Extension	5Fh

For details on overall system address mapping scheme see Section 5.1.

DOS Application Area (00000h-9FFFh)

The 640-Kbyte DOS area is divided into two parts. The 512-Kbyte area at 0 to 7FFFFh is always mapped to the main memory controlled by the GMCH, while the 128-Kbyte address range from 080000 to 09FFFFh can be mapped to PCI0 or to main SDRAM. By default, this range is mapped to main memory and can be declared as a main memory hole (accesses forwarded to PCI0) via GMCH's FDHC configuration register.

Video Buffer Area (A0000h-BFFFFh)

This 128-Kbyte area is not controlled by attribute bits. The host-initiated cycles in this region are always forwarded to either PCI0 or PCI2 unless this range is accessed in SMM mode. ***Routing of accesses is controlled by the legacy VGA control mechanism of the "virtual" PCI-PCI bridge device embedded within the GMCH***.

This area can be programmed as SMM area via the SMRAM register. When used as an SMM space, this range cannot be accessed from the hub interface.

Expansion Area (C0000h-DFFFFh)

This 128-Kbyte area is divided into eight 16-Kbyte segments that can be assigned different attributes via PAM control register as defined in Table 36.

Extended System BIOS Area (E0000h-EFFFFh)

This 64-Kbyte area is divided into four 16-Kbyte segments that can be assigned different attributes via PAM control register as defined in Table 36.

System BIOS Area (F0000h-FFFFFh)

This area is a single 64-Kbyte segment that can be assigned different attributes via PAM control register as defined in Table 36.

4.5.1.19 DRB(7:0) - DRAM Row Boundary Register - Dev #0

Row Boundary Register defines the upper boundary address of each SDRAM row in 32-Mbyte granularity.

Each row has its own DRB register. Contents of these 8-bit registers represent the boundary address in 32-Mbyte granularity. For example, a value of one indicates 32 Mbytes.

Row0:60h

Row1:61h

Row2:62h

Row3:63h

Row4:64h: Reserved

Row5:65h: Reserved

Row6:66h: Reserved

Row7:67h: Reserved

DRB0 = Total memory in row0 (in 32 Mbytes)

DRB1 = Total memory in row0 + row1 (in 32 Mbytes)

DRB4 = Total memory in row0 + row1 + row2 + row3 + (in 32 Mbytes)

The number of DRB registers and number of bits per DRB register are system dependent. For example, a system that supports four rows of SDRAM and a max memory of 1.0 Gbyte needs only four DRB registers and four bits per DRB.

GMCH supports four physical rows of Single data rate SDRAM in 2 DIMMs. The width of a row is 64 bits. Each DIMM/Row is represented by a byte. Each byte has the following format.

GMCH supported maximum memory size: 1.0 Gbyte.

Table 37. DRB(7:0) - DRAM Row Boundary Register - Dev #0

Offset:	60-67h	***Attribute:***	Read/Write (Read-Only if D_LCK = 1)
Default Value:	00h	***Size:***	8-bit

Bits	Description
7:0	**SDRAM Row Boundary Address**: This 8-bit value defines the upper and lower addresses for each SDRAM row. Bits 6:0 of this field are compared against the address lines A [31:25] to determine the upper address limit of a particular row. Bit 7 must be zero. Default value=0000/0000.

4.5.1.20 DRA(1:0) - DRAM Row Attribute Register - Dev #0

Row attribute register defines the page size of each row.

Table 38. DRA(1:0) - DRAM Row Attribute Register - Dev #0

Offset:	70-71h	***Attribute:***	Read/Write (Read-Only if D_LCK = 1)
Default Value:	FFh	***Size:***	8 bit
Row0, 1:	70h	**Row2, 3:**	71h

7	6	4	3	2	0
R	Row attribute for Row 1		R	Row attribute for Row 0	

7	6	4	3	2	0
R	Row attribute for Row 3		R	Row attribute for Row 2	

Bit	Description
3:0(7:4)	**Row Attribute**: This 4-bit field defines the page size of the row. Page size is dependent on the technology as shown below. **Bits 3:0 Page Size** “0000” 2 Kbytes “0001” 4 Kbytes “0010” 8 Kbytes “0011” 16 Kbytes “1111” Empty Row **All other combinations are reserved**. Default value=1111.

4.5.1.21 DRT - DRAM Timing Register - Dev #0

This register controls the timing of the SDRAM controller.

Table 39. DRT - DRAM Timing Register - Dev #0 (Sheet 1 of 2)

Offset:	78-7Bh	***Attribute:***	Read/Write
Default Value:	00000010h	***Size:***	32 bit

Bits	Description
31:19	**Reserved**
18:16	**DRAM Idle Timer:** This field determines the number of clock cycles that a row in the idle state (un-accessed) can allow before pre-charging all pages in that row, or powering down the row with respect to the values of bit 28 and bit 14 of DRC. Bit[18:16] Idle Clocks Before Action **0 0 0** Infinite (Counter is disabled and no action is taken) **0 0 1** 0 (Not supported on GMCH as this setting requires auto precharge) **0 1 0** 8 **0 1 1** 16 **1 0 0** 64 **1 0 1** 256 **1 1 0** 512 **1 1 1** 1024 DRC 28 (Pwr Dwn Enbl) DRC14 (Page Cls Enbl) Action on counter expiration. 0 0 None (Counter disabled) 0 1 Pre-charge all 1 0 Power down and deassert CKE, pages open. 1 1 Pre-charge all, power down and deassert CKE Default value=000. Recommended settings for DRC 28=1, DRC 14=1 and DRT 18:16 =010.
15:11	**Reserved**
10	**Activate to Precharge Delay (tRAS).** This bit controls the number of CLKs for tRAS. **0** = tRAS = 7 CLKs **1** = tRAS = 5 CLKs. Default value=0.
9:6	**Reserved**
5:4	**CAS# Latency (tCL).** This bit controls the number of CLKs between when a read command is sampled by the SDRAM and when GMCH samples read data from the SDRAM. **00** = **Reserved** **01** = 3 **10** = 2 **11** = **Reserved** Default value=01.
3	**Reserved**

Table 39. DRT - DRAM Timing Register - Dev #0 (Sheet 2 of 2)

Offset:	78-7Bh	***Attribute:***	Read/Write
Default Value:	00000010h	***Size:***	32 bit

Bits	Description
2	**DRAM RAS# to CAS# Delay (tRCD).** This bit controls the number of CLKs from a row activate command to a read or write command. **0** = 3 clocks are inserted between a row activate command and either a read or write command. **1** = 2 clocks are inserted between a row activate command and either a read or write command. Default value=0.
1	**Reserved**
0	**DRAM RAS# Precharge (tRP).** This bit controls the number of CLKs for RAS# pre-charge. **0** = 3 clocks of RAS# pre-charge are provided. **1** = 2 clocks of RAS# pre-charge are provided Default value=0.

4.5.1.22 DRC - DRAM Controller Mode Register - Dev #0

Table 40. DRC - DRAM Controller Mode Register - Dev #0 (Sheet 1 of 3)

Offset:	7C-7Fh	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32-bit

Bits	Description
31:30	**Specification Revision Number.** Hardwired to "**00**" on GMCH.
29	**Initialization Complete (IC):** Setting this bit to one enables SDRAM refreshes. On power up and S3 exit, the BIOS initializes the SDRAM array and sets this bit to one. This bit works in combination with the RMS bits in controlling refresh state: **IC** **RMS** **Refresh State** 0 XXX OFF X 000 OFF 1 001 ON 1 010 ON 1 011 ON 1 111 ON Default value=0.
28	**DRAM Row Power- Mgmt Enable:** When this bit is set to one, a SDRAM row is powered down (issued a power down command and SMCKE deasserted) after the SDRAM idle timer (as programmed in DRT) expires. During a refresh, rows in the low power state are powered up and refreshed. Coming out of a refresh, all rows are powered up. Default value=0.
27	**Reserved**.

Table 40. DRC - DRAM Controller Mode Register - Dev #0 (Sheet 2 of 3)

Offset:	7C-7Fh	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32-bit

Bits	Description
26:24	**Active Row Count:** This field determines the number of rows the SDRAM controller allows in the active state if SDRAM row power management is enabled (bit 28). All populated rows not in the active state are in power down. An access to a row in power down causes that row to exit power down and the LRU row is placed into power down if the number of active rows is greater than that allowed by this register. (See BIOS specification for the latest value. To receive the BIOS specification, contact your Intel Field Representative). **Bit[26:24] Maximum number of Active Rows** **0 0 0** All rows allowed to be in active state. **0 0 1** One Row **0 1 0** Two Rows **0 1 1** Three Rows **1 0 0** Four Rows **1 0 1** Reserved **1 1 0** Reserved **1 1 1** Reserved Default value=000.
23:20	**Reserved**
19:15	**Reserved**
14	**Page Close Enable:** When this bit is set to one, SDRAM row pages are closed after the SDRAM idle timer (as programmed in DRT) expires. Default value=0.
13:11	**Reserved**
10:8	**Refresh Mode Select (RMS)**: Bits determine if refresh is enabled and refresh rate. 000: Refresh Disabled. 001: Refresh Enabled. Refresh interval 15.6 µs. 010: Refresh Enabled. Refresh interval 7.8 µs. 011: **Reserved** 111: Refresh Enabled. Refresh interval 128 clocks. (Fast refresh mode) **All Other Combinations are reserved**. Default value=000.
7	**Reserved**

Table 40. DRC - DRAM Controller Mode Register - Dev #0 (Sheet 3 of 3)

	Offset: 7C-7Fh	***Attribute:***	Read/Write
	Default Value: 00000000h	***Size:***	32-bit

Bits	Description
6:4	**Mode Select (SMS).** These bits select the special operational mode of the GMCH SDRAM interface. The special modes are intended for initialization at power up. **000** = Self Refresh (default): In this mode, SMCKEs are deasserted. All other values cause SMCKE assertion. The exception is in C3/S1/S3 this register is programmed to "normal operation," the DRAMs are self-refreshed, and SMCKEs are deasserted. **001** = NOP Command Enable: In this mode all CPU cycles to SDRAM result in a NOP command on the SDRAM interface. **010** = All Banks Pre-charge Enable: In this mode all CPU cycles to SDRAM result in an all banks pre-charge command on the SDRAM interface. **011** = Mode Register Set Enable: In this mode all CPU cycles to SDRAM result in a mode register set command on the SDRAM interface. The command is driven on the MA[12:0] lines. MA[2:0] must always be driven to 010 for burst of 4 mode. MA3 must be driven to 1 for interleave wrap type. MA[6:4] needs to be driven based on the value programmed in the CAS# latency field. CAS LatencyMA[6:4] 2 Clocks010 3 Clocks011 MA[12:7] must be driven to 00000. BIOS must calculate and drive the correct host address for each row of memory such that the correct command is driven on the MA[12:0] lines. **100** = Reserved. **101** = Reserved. **110** = CBR Refresh Enable. In this mode all CPU cycles to SDRAM result in a CBR cycle on the SDRAM interface. **111** = Normal Operation. Default value=000.
3:2	**Reserved**
1:0	**Reserved**

4.5.1.23 DTC - DRAM Throttling Control Register - Dev #0

Throttling is independent for read and write operations. If the number of oct-words (16 bytes) read or written during this window exceeds the DRAM bandwidth threshold defined below, then the DRAM throttling mechanism is invoked to limit DRAM reads/writes to a lower bandwidth checked over smaller time windows. The throttling is active for the remainder of the current GDWS and for the next GDSW after which it returns to non-throttling mode. The throttling mechanism accounts for the actual bandwidth consumed during the sampling window, by reducing the allowed bandwidth within the smaller throttling window based on the bandwidth consumed during the sampling period.

Table 41. DTC - DRAM Throttling Control Register - Dev #0 (Sheet 1 of 2)

	Offset: 8C-8Fh	***Attribute:***	Read/Write/Lock
	Default Value: 00000000h	***Size:***	32-bit

Bits	Description
31	**Throttle Lock (TLOCK):** This bit secures the SDRAM throttling control register. After one is written to this bit, all of the configuration register bits in DTC (including TLOCK) documented below become read-only. Default value=0.
30	**Intel Reserved**
29:28	DRAM Throttle Mode (TMODE): Bits Mode 0 0 Throttling turned off. 0 1 Bandwidth counter mechanism is enabled. When bandwidth exceeds threshold set in the r/w PTC field, DRAM read/write throttling begins. 1 0 Thermal sensor based throttling enabled. When the device's thermal sensor is tripped DRAM write throttling begins based on settings programmed in WPTC. Read throttling is disabled. 1 1 With this setting thermal sensor and DRAM counter mechanisms are both enabled. However, read throttling is triggered by bandwidth-counter only while write throttling is triggered by thermal sensor or counter. Both read and write throttling mechanisms use programmed values in the throttle control registers. Default value=00
27:26	**Reserved**
25:24	**Read Power Throttle Control.** These bits select the power throttle bandwidth limits for read operations to system memory. **Read/write, read-only if throttle locked.** **Bits Power throttle bandwidth limit** 00 No Limit (Default) 01 Limit at 65% 10 Limit at 55% 11 Limit at 45% Default value=00
23:22	**Reserved**
21:20	**Write Power Throttle Control.** These bits select the power throttle bandwidth limits for write operations to system memory. **Read/write, read-only if throttle is locked.** **Bits Power throttle bandwidth limit** 00 No Limit (Default) 01 Limit at 65% 10 Limit at 55% 11 Limit at 45% Default value=00

Table 41. DTC - DRAM Throttling Control Register - Dev #0 (Sheet 2 of 2)

Offset:	8C-8Fh	*Attribute:*	Read/Write/Lock
Default Value:	00000000h	*Size:*	32-bit

Bits	Description
19:16	**Reserved**
15:8	**Global DRAM Sampling Window (GDSW)**: This 8-bit value is multiplied by four to define the length of time in milliseconds (0-1020) over which the number of oct-words (16 bytes) read/written is counted and throttling is imposed. Default value=00000000.
7:0	**Reserved**

4.5.1.24 SMRAM - System Management RAM Control Register - Dev #0

The SMRAM register controls how accesses to compatible and extended SMRAM spaces are treated. The open, close, and lock bits function only when G_SMRAME bit is set to one. The open bit must be reset before the lock bit is set.

Table 42. SMRAM - System Management RAM Control Register - Dev #0 (Sheet 1 of 2)

Offset:	90h	*Attribute:*	Read/Write/Lock, Read-Only
Default Value:	02h	*Size:*	8-bit

Bits	Description
7	**Reserved**
6	**SMM Space Open (D_OPEN)**: When D_OPEN=1 and D_LCK=0, the SMM space SDRAM is made visible even when SMM decode is not active. This is intended to help BIOS initialize SMM space. Software should ensure that D_OPEN=1 and D_CLS=1 are not set at the same time. When D_LCK is set to one, D_OPEN is reset to zero and becomes read-only. Default value=0.
5	**SMM Space Closed (D_CLS)**: When D_CLS = 1 SMM space DRAM is not accessible to data references, even if SMM decode is active. Code references may still access SMM space SDRAM. This allows SMM software to reference “through” SMM space to update the display even when SMM is mapped over the VGA range. Software should ensure that D_OPEN=1 and D_CLS=1 are not set at the same time. Default value=0.

Table 42. SMRAM - System Management RAM Control Register - Dev #0 (Sheet 2 of 2)

	Offset: 90h	***Attribute:***	Read/Write/Lock, Read-Only
	Default Value: 02h	***Size:***	8-bit

Bits	Description
4	**SMM Space Locked (D_LCK)**: When D_LCK is set to one then D_OPEN is reset to zero and D_LCK, D_OPEN, G_SMRAME, C_BASE_SEG, GMS, DRB, DRA, H_SMRAM_EN, TSEG_SZ and TSEG_EN become read-only. GBA[15:0] and GAR[15:0] associated with the SDRAM controller also become read-only after D_LCK is set. D_LCK can be set to one via a normal configuration space write but can only be cleared by a full reset. The combination of D_LCK and D_OPEN provide convenience with security. The BIOS can use the D_OPEN function to initialize SMM space and then use D_LCK to “lock down” SMM space in the future so that no application software (or BIOS itself) can violate the integrity of SMM space, even if the program has knowledge of the D_OPEN function. Default value=0.
3	**Global SMRAM Enable (G_SMRAME)**. If set to one, then compatible SMRAM functions are enabled, providing 128 Kbytes of SDRAM accessible at the A0000h address while in SMM (ADS# with SMM decode). To enable extended SMRAM function this bit has be set to one. Refer to the section on SMM for more details. After D_LCK is set, this bit becomes read-only. Default value=0.
2:0	**Compatible SMM Space Base Segment (C_BASE_SEG) (RO)**. This field indicates the location of SMM space. “SMM DRAM” is not remapped. It is simply “made visible” if the conditions are right to access SMM space, otherwise the access is forwarded to hub interface. C_BASE_SEG is hardwired to 010 to indicate that the GMCH-M supports the SMM space at A0000h-BFFFFh. Default value=010.

4.5.1.25 ESMRAMC - Extended System Management RAM Control Register - Dev #0

The extended SMRAM register controls the configuration of extended SMRAM space. The extended SMRAM (E_SMRAM) memory provides a writeback cacheable SMRAM memory space that is above 1 Mbyte.

Table 43. ESMRAMC - Extended System Management RAM Control Register - Dev #0

Offset:	91h	***Attribute:***	Read/Write
Default Value:	38h	***Size:***	8-bit

Bits	Description
7	**H_SMRAM_EN (H_SMRAME):** Controls the SMM memory space location (i.e., above 1 Mbyte or below 1 Mbyte). When G_SMRAME is one this bit is set to one, the high SMRAM memory space is enabled. SMRAM accesses from 0FEDA0000h to 0FEDBFFFFh are remapped to SDRAM address 000A0000h to 000BFFFFh. After D_LCK is set, this bit becomes read-only. Default value=0.
6	**E_SMRAM_ERR (E_SMERR):** This bit is set when CPU accesses the defined memory ranges in extended SMRAM (high memory and T-segment) while not in SMM space and with the D-OPEN bit = 0. It is the software's responsibility to clear this bit. The software must write one to this bit to clear it. Default value=0.
5	**SMRAM_Cache (SM_CACHE)**: This bit is forced to one by the GMCH. Default value=1.
4	**SMRAM_L1_EN (SM_L1):** This bit is forced to one by the GMCH. Default value=1.
3	**SMRAM_L2_EN (SM_L2):** This bit is forced to one by the GMCH. Default value=1.
2	**Reserved**
1	**TSEG_SZ(T_SZ):** Selects the size of the TSEG memory block if enabled. This memory is taken from the top of SDRAM space (i.e,. TOM - TSEG_SZ), which is no longer claimed by the memory controller. This field decodes as follows: **TSEG_SZ** **Description** 0 (TOM-512K) to TOM 1 (TOM-1M) to TOM **After D_LCK is set, this bit becomes read-only**. Default value=0.
0	**TSEG_EN (T_EN):** Enabling of SMRAM memory (TSEG, 512 Kbytes or 1 Mbyte of additional SMRAM memory) for extended SMRAM space only. When G_SMRAME =1 and TSEG_EN = 1, the TSEG is enabled to appear in the appropriate physical address space. **After D_LCK is set, this bit becomes read-only**. Default value=0.

4.5.1.26 ERRSTS - Error Status Register - Dev #0

This register is used to report various error conditions via hub interface special cycles. A SERR, SMI, or SCI error hub interface special cycle may be generated on a zero-to-one transition of any of these flags when enabled in the PCICMD/ERRCMD, SMICMD, or SCICMD registers respectively.

Table 44. ERRSTS - Error Status Register - Dev #0

Offset:	92-93h	***Attribute:***	Read/Write Clear
Default Value:	0000h	***Size:***	16-bit

Bits	Description
15:13	**Reserved**
12	GMCH Software-Generated Event for SMI. This indicates the source of the SMI was a device #2 software event for the local memory interface. Software must write one to clear this bit. (Local memory no longer supported.)
11	**Reserved**
10	**Reserved**
9	**LOCK to non-DRAM Memory Flag (LCKF).** (R/WC) When this bit is set it indicates that a CPU-initiated LOCK cycle targeting non-DRAM memory space occurred. Software must write one to clear this status bit.
8	Received Refresh Time-out. This bit is set when 1024 memory core refresh is queued up. Software must write one to clear this status bit.
7	**DRAM Throttle Flag (DTF) (R/WC).** When this bit is set it indicates that the DRAM throttling condition occurred. Software must write one to clear this status bit.
6	**Reserved**
5	**Received Unimplemented Special Cycle Hub Interface Completion Packet FLAG (UNSC) (R/WC).** When this bit is set, it indicates that the GMCH initiated a hub interface request that was terminated with an unimplemented special cycle completion packet. Software must write one to clear this status bit.
4	**Reserved**
3	**Reserved**
2	**Reserved**
1:0	**Reserved**

4.5.1.27 ERRCMD - Error Command Register - Dev #0

This register enables various errors to generate a SERR hub interface special cycle. Because the GMCH does not have a SERR# signal, SERR messages are passed from the GMCH to the ICH4 over the hub interface. The actual generation of the SERR message is globally enabled for device #0 via the PCI command register.

An error can generate one and only one hub interface error special cycle. The software is responsible to ensure that when a SERR error message is enabled for an error condition, SMI and SCI error messages are disabled for that same error condition.

Table 45. ERRCMD - Error Command Register - Dev #0

Offset: 94-95h	***Attribute:*** Read/Write
Default Value: 0000h	***Size:*** 16-bit

Bits	Description
15:10	**Reserved**
9	**SERR on LOCK to Non-SDRAM Memory**. When this bit is set to one, the GMCH generates a SERR hub interface special cycle when a CPU initiated LOCK transaction targeting non-SDRAM memory space occurs. If this bit is zero then reporting this condition is disabled. Default value=0.
8	**SERR on SDRAM Refresh time-out**. When this bit is set to one, the GMCH generates a SERR hub interface special cycle when a SDRAM refresh time-out occurs. If this bit is zero then reporting this condition is disabled. Default value=0.
7	**SERR on SDRAM Throttle Condition**. When this bit is set to one, the GMCH generates a SERR hub interface special cycle when a SDRAM read or write throttle condition occurs. If this bit is zero then reporting this condition is disabled. Default value=0.
6	**SERR on Receiving Target Abort on Hub Interface**. When this bit is set to one, the GMCH generates a SERR hub interface special cycle when a GMCH originated hub interface cycle is terminated with a target abort. If this bit is zero then reporting this condition is disabled. Default value=0.
5	**SERR on Receiving Unimplemented Special Cycle Hub Interface Completion Packet**. When this bit is set to one, the GMCH generates a SERR hub interface special cycle when a GMCH initiated hub interface request is terminated with a unimplemented special cycle completion packet. If this bit is zero then reporting this condition is disabled. Default value=0.
4	**Reserved**
3	**Reserved**
2	**Reserved**
1:0	**Reserved**

Table 46. Summary of GMCH Error Sources, Enables and Status Flags (Sheet 1 of 2)

Error Event	Hub I/F Message	Enable Bits Required to be Set	Status Flags Set
SDRAM Refresh Time-out	SERR	PCICMD bit 8 ERRCMD bit 8	PCISTS bit 14 ERRSTS bit 8
CPU LOCK to non-SDRAM Memory	SERR	PCICMD bit 8 ERRCMD bit 9	PCISTS bit 14 ERRSTS bit 9

Table 46. Summary of GMCH Error Sources, Enables and Status Flags (Sheet 2 of 2)

SDRAM Throttle	SERR	PCICMD bit 8 ERRCMD bit 7	PCISTS bit 14 ERRSTS bit 7
Received Hub Interface Target Abort	SERR	PCICMD bit 8 ERRCMD bit 6	PCISTS bit 14 PCISTS bit 12
Unimplemented Special Cycle	SERR	PCICMD bit 8 ERRCMD bit 5	PCISTS bit 14 ERRSTS bit 5

4.5.1.28 ACAPID - AGP Capability Identifier Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 47. ACAPID - AGP Capability Identifier Register - Dev #0

Offset:	A0-A3h	***Attribute:***	Read-Only
Default Value:	00200002h	***Size:***	32-bit

4.5.1.29 AGPSTAT - AGP Status Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 48. AGPSTAT - AGP Status Register - Dev #0

Offset:	A4-A7h	***Attribute:***	Read-Only
Default Value:	1F000217h	***Size:***	32-bit

4.5.1.30 AGPCMD - AGP Command Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 49. AGPCMD - AGP Command Register - Dev #0

Offset:	A8-ABh	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32-bit

4.5.1.31 AGPCTRL - AGP Control Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 50. AGPCTRL - AGP Control Register - Dev #0

Offset:	B0-B1h	*Attribute:*	Read/Write
Default Value:	00000000h	*Size:*	32-bit

4.5.1.32 AFT - AGP Functional Test Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 51. AFT - AGP Functional Test Register - Dev #0

Offset:	B2-B3h	*Attribute:*	Read/Write
Default Value:	0000h	*Size:*	16-bit

4.5.1.33 APSIZE - Aperture Size - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 52. APSIZE - Aperture Size - Dev #0

Offset:	B4h	*Attribute:*	Read/Write
Default Value:	00h	*Size:*	8-bit

4.5.1.34 ATTBASE - Aperture Translation Table Base Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 53. ATTBASE - Aperture Translation Table Base Register - Dev #0

Offset:	B8-BBh	*Attribute:*	Read/Write
Default Value:	00000000h	*Size:*	32-bit

4.5.1.35 AMTT - AGP Interface Multi-Transaction Timer Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 54. AMTT - AGP Interface Multi-Transaction Timer Register - Dev #0

Offset:	BCh	*Attribute:*	Read/Write
Default Value:	00h	*Size:*	8-bit

4.5.1.36 LPTT - Low Priority Transaction Timer Register - Dev #0

This register has no functionality in the Intel 835 chipset.

Table 55. LPTT - Low Priority Transaction Timer Register - Dev #0

Offset:	BDh	***Attribute:***	Read/Write
Default Value:	00h	***Size:***	8-bit

4.5.1.37 BUFF_SC - System Memory Buffer Strength Control Register - Dev #0

Table 56. BUFF_SC - System Memory Buffer Strength Control Register - Dev #0

Offset:	EC-EFh	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32-bit

4.5.1.37.1 SDR Drive Strength Register Description

The system memory buffer strength control register programs drive strengths and slew rate and for each buffer category based on loading detected by SPD. SMCS, SMCKE, and CMCLK buffers have independent control for each DIMM and are programmed to the same strength for front- and backside of each DIMM. If the BIOS detects different loading on the backside of the DIMM (i.e., 96 Mbytes), it should ignore the devices on the backside of the DIMM.

Table 57. SDR Drive Strength Register (Sheet 1 of 3)

Bit	Descriptions
31	**Reserved**
30	**SMCLK [3:2] Slew Rate**. This field sets the slew rate of the SMCLK [3:2] pins. 0 = Normal slew rate. 1 = Fast slew rate for reduced Tco. Default value=0.
29	**SMCLK [1:0] Slew Rate**. This field sets the slew rate of the SMCLK [1:0] pins. 0 = Normal slew rate. 1 = Fast slew rate for reduced Tco. Default value=0.
28	**Reserved**
27	**SMCS [3:2], SMCKE [3:2] Slew Rate**. This field sets the slew rate of the SMCS [3:2], SMCKE [3:2] pins. 0 = Normal slew rate. 1 = Fast slew rate for reduced Tco. Default value=0.
26	**SMCS [1:0], SMCKE [1:0] Slew Rate**. This field sets the slew rate of the SMCS [1:0], SMCKE [1:0] pins. 0 = Normal slew rate. 1 = Fast slew rate for reduced Tco. Default value=0.

Table 57. SDR Drive Strength Register (Sheet 2 of 3)

Bit	Descriptions
25	**SMDQ [63:0], SMDQM [7:0] Slew Rate**. This field sets the slew rate of the SMDQ [63:0], SMDQM [7:0] pins. 0 = Normal slew rate. 1 = Fast slew rate for reduced Tco. Default value=0.
24	**SMMA [12:0], SMBA[1:0], SMRAS, SMCAS, SMWE Slew Rate**. This field sets the slew rate of the SMMA [12:0], SMBA [1:0], SMRAS, SMCAS, and SMWE pins. 0 = Normal slew rate. 1 = Fast slew rate for reduced Tco. Default value=0.
23:21	**Reserved**
20:18	**SMCLK [3:2] Buffer Strength**. This field sets the buffer strength of the SMCLK [3:2] pins. 000 = 0.75X 001 = 1X 010 = 1.25X 011 = 1.5X 100 = 2X 101 = 2.5X 110 = 3X 111 = 4X Default value=000.
17:15	**SMCLK [1:0] Buffer Strength**. This field sets the buffer strength of the SMCLK [1:0] pins. 000 = 0.75X 001 = 1X 010 = 1.25X 011 = 1.5X 100 = 2X 101 = 2.5X 110 = 3X 111 = 4X Default value=000.
14:12	**Reserved**
11:9	**SMCS [3:2], SMCKE [3:2] Buffer Strength**. This field sets the buffer strength of the SMCS [3:2], SMCKE [3:2] pins. 000 = 0.75X 001 = 1X 010 = 1.25X 011 = 1.5X 100 = 2X 101 = 2.5X 110 = 3X 111 = invalid Default value=000.

Table 57. SDR Drive Strength Register (Sheet 3 of 3)

Bit	Descriptions
8:6	**SMCS [1:0], SMCKE[1:0] Buffer Strength**. This field sets the buffer strength of the SMCS [1:0], SMCKE [1:0] pins. 000 = 0.75X 001 = 1X 010 = 1.25X 011 = 1.5X 100 = 2X 101 = 2.5X 110 = 3X 111 = invalid Default value=000.
5:3	**SMDQ [63:0], SMDQM [7:0] Buffer Strength**. This field sets the buffer strength of the SMDQ [63:0], SMDQM [7:0] pins. 000 = 0.75X 001 = 1X 010 = 1.25X 011 = 1.5X 100 = 2X 101 = 2.5X 110 = 3X 111 = invalid Default value=000.
2:0	**SMMA [12:0], SMBA[1:0], SMRAS, SMCAS, SMWE Buffer Strength**. This field sets the buffer strength of the SMMA [12:0], SMBA [1:0], SMRA, SMCAS, and SMWE pins. 000 = 0.75X 001 = 1X 010 = 1.25X 011 = 1.5X 100 = 2X 101 = 2.5X 110 = 3X 111 = invalid Default value=000.

4.5.2 Intel® 835 Chipset Integrated Graphics Device Registers – Dev #2

This section contains the PCI configuration registers listed in order of ascending offset address. Device #2 incorporates two functions, #0 and #1.

Table 58. Integrated Graphics Device Configuration Space (Device #2) (Sheet 1 of 2)

Address Offset	Register Symbol	Register Name	Default Value Function #0	Default Value Function #1	Access	Regs in Func #1
00-01h	VID2	Vendor Identification	8086h	8086h	RO	COF0
02-03h	DID2	Device Identification	357Bh	357Bh	RO	COF0
04-05h	PCICMD2	PCI Command Register	0000h	0000h	RO,R/W	UIF1
06-07h	PCISTS2	PCI Status Register	0090h	0090h	RO,R/WC	UIF1
08h	RID2	Revision Identification	00h	00h	RO	COF0
09-0Bh	CC	Class Code	030000h	038000h	RO	UIF1
0Ch	CLS	Cache Line Size Register	00h	00h	RO	COF0
0Dh	MLT2	Master Latency Timer	00h	00h	RO	COF0
0Eh	HDR2	Header Type	00h	00h	RO	UIF1
0Fh	-	Intel Reserved	-	-	-	-
10-13h	GMADR	Graphics Memory Range Address	00000008h	00000008h	RO,R/W	UIF1
14-17h	MMADR	Memory Mapped Range Address	00000000h	00000000h	RO,R/W	UIF1
18-2Bh	-	Intel Reserved	-	-	-	-
2C-2Dh	SVID2	Subsystem Vendor Identification	0000h	0000h	R/WO in F# 0	COF0
2E-2Fh	SID2	Subsystem Identification	0000h	0000h	R/WO in F# 0	COF0
30-33h	ROMADR	Video BIOS ROM Base Address	00000000h	00000000h	RO,R/W	COF0
34h	CAPPOINT	Capabilities Pointer	D0h	D0h	RO	COF0
35-3Bh	-	Intel Reserved	-	-	-	-
3Ch	INTRLINE	Interrupt Line Register	00h	00h	R/W, RO in F# 1	-
3Dh	INTRPIN	Interrupt Pin Register	01h	00h	RO, Reserved In F #1	-
3Eh	MINGNT	Minimum Grant Register	00h	00h	RO	COF0
3Fh	MAXLAT	Maximum Latency Register	00h	00h	RO	COF0
40-CFh	-	Intel Reserved	-	-	-	-
D0-D1h	PMCAPID	Power Management Capabilities ID	0001h	0001h	RO	COF0

Table 58. Integrated Graphics Device Configuration Space (Device #2) (Sheet 2 of 2)

Address Offset	Register Symbol	Register Name	Default Value Function #0	Default Value Function #1	Access	Regs in Func #1
D2-D3h	PMCAP	Power Management Capabilities	0221h	0221h	RO	COF0
D4-D5h	PMCS	Power Management Control/Status	0000h	0000h	RO,R/W	UIF1
D6-FFh	-	Intel Reserved	-	-	-	-

NOTES:
1. COF0: Copy of Function #0. No hardware implemented for this register in function #1.
2. UIF1: Unique in Function #1. Hardware is implemented for this register in function #1, and may be RO or R/W.

4.5.2.1 VID2 - Vendor Identification Register – Dev #2

The VID Register contains the vendor identification number. This 16-bit register combined with the device identification register uniquely identifies any PCI device. Writes to this register have no effect.

Table 59. VID2 - Vendor Identification Register – Dev #2

Offset:	00h-01h	***Attribute:***	Read-Only
Default Value:	8086h	***Size:***	16-bit

Bits	Description
15:0	**Vendor Identification Number**. This is a 16-bit value assigned to Intel. Default value=1000/0000/1000/0110.

4.5.2.2 DID2 - Device Identification Register - Dev #2

This 16-bit register combined with the vendor identification register uniquely identifies any PCI device. Writes to this register have no effect.

Table 60. DID2 - Device Identification Register - Dev #2

Offset:	02h-03h	***Attribute:***	Read-Only
Default Value:	357Bh	***Size:***	16-bit

Bits	Description
15:0	**Device Identification Number**. This is a 16 bit value assigned to the GMCH IGD. Default value=0011/0101/0111/1011.

4.5.2.3 PCICMD2 - PCI Command Register - Dev #2

This 16-bit register provides basic control over the IGD's ability to respond to PCI cycles. The PCICMD register in the IGD disables the IGD PCI-compliant master accesses to main memory.

Table 61. PCICMD2 - PCI Command Register - Dev #2

	Offset: 04h-05h	***Attribute:***	Read-Only, Read/Write
	Default Value: 0000h	***Size:***	16-bit

Bits	Description
15:10	**Reserved**
9	**Fast Back-to-Back (FB2B) - RO**. (Not Implemented). Hardwired to zero.
8	**SERR# Enable (SERRE) - RO**. (Not Implemented). Hardwired to zero.
7	**Address/Data Stepping - RO**. (Not Implemented). Hardwired to zero.
6	**Parity Error Enable (PERRE) - RO**. (Not Implemented). Hardwired to zero. Because the IGD belongs to the category of devices that does not corrupt programs or data in system memory or hard drives, the IGD ignores any parity error that it detects and continues with normal operation.
5	**Video Palette Snooping (VPS) - RO**. This bit is hardwired to zero to disable snooping.
4	**Memory Write and Invalidate Enable (MWIE) - RO**. Hardwired to zero. The IGD does not support memory write and invalidate commands.
3	**Special Cycle Enable (SCE) - RO**. This bit is hardwired to zero. The IGD ignores special cycles.
2	**Bus Master Enable (BME) - R/W**. Set to one to enable the IGD to function as a PCI compliant master. Set to zero to disable IGD bus mastering. Default value=0.
1	**Memory Access Enable (MAE) - R/W**. This bit controls the IGD's response to memory space accesses. 0 = Disable 1 = Enable Default value=0.
0	**I/O Access Enable (IOAE) - R/W**. This bit controls the IGD's response to I/O space accesses. 0 = Disable 1 = Enable. Default value=0.

4.5.2.4 PCISTS2 - PCI Status Register - Dev #2

PCISTS is a 16-bit status register that reports the occurrence of a PCI compliant master abort and PCI compliant target abort. PCISTS also indicates the DEVSEL# timing that has been set by the IGD.

Table 62. PCISTS2 - PCI Status Register - Dev #2

Offset:	06h-07h	*Attribute:*	Read-Only, Read/Write Clear
Default Value:	0090h	*Size:*	16-bit

Bits	Description
15	**Detected Parity Error (DPE) - RO**. Since the IGD does not detect parity, this bit is always set to zero. Default value=0.
14	**Signaled System Error (SSE) - R/WC**. The IGD never asserts SERR#, therefore this bit is hardwired to zero.
13	**Received Master Abort Status (RMAS) - R/WC**. The IGD never gets a master abort, therefore this bit is hardwired to zero.
12	**Received Target Abort Status (RTAS) - R/WC**. The IGD never gets a target abort, therefore this bit is hardwired to zero.
11	**Signaled Target Abort Status (STAS)**. Hardwired to zero. The IGD does not use target abort semantics.
10:9	**DEVSEL# Timing (DEVT) - RO**. NA - Hardwired to 00.
8	**Data Parity Detected (DPD) - R/WC**. Because parity error response is hardwired to disabled (and the IGD does not do any parity detection), this bit is hardwired to zero.
7	**Fast Back-to-Back (FB2B). Hardwired to 1**. The IGD accepts fast back-to-back when the transactions are not to the same agent.
6	**User Defined Format (UDF).** Hardwired to zero.
5	**66 MHz PCI Capable (66C).** NA - Hardwired to zero.
4	**CAP LIST - RO**. This bit is set to one to indicate that the register at 34h provides an offset into the function's PCI configuration space containing a pointer to the location of the first item in the list. Default value=1.
3:0	**Reserved**

4.5.2.5 RID2 - Revision Identification Register - Dev #2

This register contains the revision number of the IGD. These bits are read-only; writes to this register have no effect.

Table 63. RID2 - Revision Identification Register - Dev #2

Offset:	08h	*Attribute:*	Read-Only
Default Value:	00h (A0 silicon)	*Size:*	8-bit

Bits	Description
7:0	**Revision Identification Number**. This is an 8-bit value that indicates the revision identification number for the IGD. See the specification update for the latest silicon revision. Silicon RevisionDefault Value A0 0000/0000 (00h)

4.5.2.6 CC - Class Code Register - Dev #2

This register contains the device programming interface information related to the sub-class code and base class code definition for the IGD. This register also contains the base class code and the function sub-class in relation to the base class code.

Table 64. CC - Class Code Register - Dev #2

Offset:	09h-0Bh	***Attribute:***	Read-Only
Default Value:	Function #0– 030000h Function #1– 038000h	***Size:***	24-bit

Bits	Description
23:16	**Base Class Code (BASEC).** 03=Display controller Default value=00000011.
15:8	**Sub-Class Code (SCC)**. **Function 0:** Value based on Device #0 GCC1 Bit 1. 0: 00h= VGA compatible (not supported) 1: 80h= Non VGA. **Function 1:** 80h=Non VGA; Default value=00000000.
7:0	**Programming Interface (PI)**. 00h=Hardwired as a display controller. Default value=00000000.

4.5.2.7 CLS - Cache Line Size Register - Dev #2

The IGD does not support this register as a PCI slave.

Table 65. CLS - Cache Line Size Register - Dev #2

Offset:	0Ch	***Attribute:***	Read-Only
Default Value:	00h	***Size:***	8-bit

Bits	Description
7:0	**Cache Line Size (CLS).** This field is hardwired to zeros. The IGD as a PCI-compliant master does not use the memory write and invalidate command and, in general, does not perform operations based on cache line size. Default value=00000000.

4.5.2.8 MLT2 - Master Latency Timer Register - Dev #2

The IGD does not support the programmability of the master latency timer because it does not perform bursts.

Table 66. MLT2 - Master Latency Timer Register - Dev #2

Offset: 0Dh	***Attribute:*** Read-Only
Default Value: 00h	***Size:*** 8-bit

Bits	Description
7:0	**Master Latency** Timer Count Value. Hardwired to 00000000.

4.5.2.9 HDR2 - Header Type Register - Dev #2

This register contains the header type of the IGD.

Table 67. HDR2 - Header Type Register - Dev #2

Offset: 0Eh	***Attribute:*** Read-Only
Default Value: 00h	***Size:*** 8-bit

Bits	Description
7	**Multi-Function Status (MFunc).** Indicates if the device is a multi-function device. The value of this register is determined by GCC1 bit 2. If GCC1 bit 2 is set this bit is one indicating device #2 to be multi-function. Default value=0.
6:0	**Header Code (H)**. This is an 7-bit value that indicates the header code for the IGD. This code has the value 00h, indicating a type 0 configuration space format. Default value=00000000.

4.5.2.10 GMADR - Graphics Memory Range Address Register - Dev #2

This register requests allocation for the IGD graphics memory. The allocation is for either 64 Mbytes or 128 Mbytes and the base address is defined by bits [31:27,26].

Table 68. GMADR - Graphics Memory Range Address Register - Dev #2

Offset: 10-13h	***Attribute:*** Read/Write, Read-Only
Default Value: 00000008h	***Size:*** 32-bit

Bits	Description
31:27	**Memory Base Address - R/W**. Set by the OS, these bits correspond to address signals [31:26]. Default value=00000.
26	**128-Mbyte Address Mask – RO, R/W.** The operation of this bit is controlled via device #0 register GCCR. If the signal is low this bit is read-only with a value of zero, indicating a memory range of 128 Mbytes. If the signal is high, this bit becomes R/W, indicating a memory range of 64 Mbytes (where system software programs the bit to the appropriate address bit value. Default value=0.
25:4	**Address Mask - RO**. Hardwired to zeros to indicate (at least) a 32-Mbyte address range.

Table 68. GMADR - Graphics Memory Range Address Register - Dev #2

	Offset: 10-13h	***Attribute:***	Read/Write, Read-Only
	Default Value: 00000008h	***Size:***	32-bit

Bits	Description
3	**Prefetchable Memory - RO**. Hardwired to one to enable prefetching.
2:1	**Memory Type - RO**. Hardwired to zero to indicate 32-bit address.
0	**Memory/IO Space - RO**. Hardwired to zero to indicate memory space.

4.5.2.11 MMADR - Memory Mapped Range Address Register - Dev #2

This register requests allocation for the IGD registers and instruction ports. The allocation is for 512 Kbytes and the base address is defined by bits [31:19].

Table 69. MMADR - Memory Mapped Range Address Register - Dev #2

	Offset: 14-17h	***Attribute:***	Read/Write, Read-Only
	Default Value: 00000000h	***Size:***	32-bit

Bits	Description
31:19	**Memory Base Address - R/W.** Set by the OS, these bits correspond to address signals [31:19]. Default value=0.
18:4	**Address Mask - RO**. Hardwired to zeros to indicate 512-Kbyte address range.
3	**Prefetchable Memory - RO**. Hardwired to zero to prevent prefetching.
2:1	**Memory Type - RO**. Hardwired to zeros to indicate 32-bit address.
0	**Memory/IO Space - RO.** Hardwired to zero to indicate memory space.

4.5.2.12 SVID2 - Subsystem Vendor Identification Register - Dev #2

Table 70. SVID2 - Subsystem Vendor Identification Register - Dev #2

	Offset: 2C-2Dh	***Attribute:***	Read/Write Once
	Default Value: 0000h	***Size:***	16-bit

Bits	Description
15:0	**Subsystem Vendor ID**. This value is used to identify the vendor of the subsystem. This register should be programmed by BIOS during boot-up. After it is written, this register becomes read-only. This register can be cleared only by a reset. Default value=0000000000000000.

4.5.2.13 SID2 - Subsystem Identification Register - Dev #2

Table 71. SID2 - Subsystem Identification Register - Dev #2

Offset: 2E-2Fh	*Attribute:* Read/Write Once
Default Value: 0000h	*Size:* 16-bit

Bits	Description
15:0	**Subsystem Identification**. This value is used to identify a particular subsystem. This field should be programmed by BIOS during boot-up. After it is written, this register becomes read-only. This register can be cleared only by a reset. Default value=0000000000000000.

4.5.2.14 ROMADR - Video BIOS ROM Base Address Registers - Dev #2

The IGD does not use a separate BIOS ROM, therefore this register is hardwired to zeros.

Table 72. ROMADR - Video BIOS ROM Base Address Registers - Dev #2

Offset: 30-33h	*Attribute:* Read/Write, Read-Only
Default Value: 00000000h	*Size:* 32-bit

Bits	Description
31:18	**ROM Base Address - R/W**. Hardwired to zeros.
17:11	**Address Mask - RO**. Hardwired to zeros to indicate 256-Kbyte address range.
10:1	**Reserved**. Hardwired to zeros.
0	**ROM BIOS Enable - R/W**. 0 = ROM not accessible. Default value=0.

4.5.2.15 CAPPOINT - Capabilities Pointer Register - Dev #2

Table 73. CAPPOINT - Capabilities Pointer Register - Dev #2

Offset: 34h	*Attribute:* Read-Only
Default Value: D0h	*Size:* 8-bit

Bits	Description
7:0	**Capabilities Pointer Value**. This field contains an offset into the function's PCI configuration space for the first item in the new capabilities linked list, the ACPI registers at address D0h. Default value=11010000.

4.5.2.16 INTRLINE - Interrupt Line Register - Dev #2

Table 74. INTRLINE - Interrupt Line Register - Dev #2

Offset:	3Ch	***Attribute:***	Read/Write
Default Value:	Function #0– 00h Function #1– 00h	***Size:***	8-bit

Bits	Description
7:0	**Interrupt Connection**. Used to communicate interrupt line routing information. POST software writes the routing information into this register as it initializes and configures the system. The value in this register indicates which input of the system interrupt controller to which the device's interrupt pin is connected. Default value=00000000.

4.5.2.17 NTRPIN - Interrupt Pin Register - Dev #2

Table 75. NTRPIN - Interrupt Pin Register - Dev #2

Offset:	3Dh	***Attribute:***	Read-Only
Default Value:	Function #0– 01h Function #1– 00h	***Size:***	8-bit

Bits	Description
7:0	**Interrupt Pin**. As a single function device, the IGD specifies INTA# as its interrupt pin. 01h=INTA#. For function #1, this register is set to 00h. Default value=00000001.

4.5.2.18 MINGNT - Minimum Grant Register - Dev #2

Table 76. MINGNT - Minimum Grant Register - Dev #2

Offset:	3Eh	***Attribute:***	Read-Only
Default Value:	00h	***Size:***	8-bit

Bits	Description
7:0	**Minimum Grant Value**. The IGD does not burst as a PCI-compliant master. Bits[7:0]=00h. Default value=00000000.

4.5.2.19 MAXLAT - Maximum Latency Register - Dev #2

Table 77. MAXLAT - Maximum Latency Register - Dev #2

Offset: 3Fh	***Attribute:*** Read-Only
Default Value: 00h	***Size:*** 8-bit

Bits	Description
7:0	**Maximum Latency Value.** The IGD has no specific requirements for how often it needs to access the PCI bus. Bits [7:0]=00h Default value=00000000.

4.5.2.20 PMCAPID - Power Management Capabilities ID Register - Dev #2

Table 78. PMCAPID - Power Management Capabilities ID Register - Dev #2

Offset: D0h-D1h	***Attribute:*** Read-Only
Default Value: 0001h	***Size:*** 16-bit

Bits	Description
15:8	**NEXT_PTR.** This contains a pointer to next item in capabilities list. This is the final capability in the list and must be set to 00h. Default value=00000000.
7:0	**CAP_ID.** SIG defines this ID is 01h for power management. Default value=00000001.

4.5.2.21 PMCAP - Power Management Capabilities Register - Dev #2

Table 79. PMCAP - Power Management Capabilities Register - Dev #2

Offset: D2h-D3h	***Attribute:*** Read-Only
Default Value: 0221h	***Size:*** 16-bit

Bits	Description
15:11	**PME Support**. This field indicates the power states in which the IGD may assert PME#. Hardwired to zero to indicate that the IGD does not assert the PME# signal.
10	**D2**. The D2 power management state is not supported. This bit is hardwired to zero.
9	**D1**. Hardwired to one to indicate that the D1 power management state is supported.
8:6	**Reserved**. Read as zeros.
5	**Device Specific Initialization (DSI).** Hardwired to one to indicate that special initialization of the IGD is required before generic class device driver is to use it.
4	**Auxiliary Power Source**. Hardwired to zero.
3	**PME Clock**. Hardwired to zero to indicate IGD does not support PME# generation.
2:0	**Version.** Hardwired to 001b to indicate there are 4 bytes of power management registers implemented.

4.5.2.22 PMCS - Power Management Control/Status Register - Dev #2

Table 80. PMCS - Power Management Control/Status Register - Dev #2

	Offset: D4h-D5h	***Attribute:***	Read/Write, Read-Only
	Default Value: 0000h	***Size:***	16-bit

Bits	Description
15	**PME_Status - RO**. This bit is zero to indicate that IGD does not support PME# generation from D3 (cold). Default value=0.
14:13	**Reserved** Default value=0.
12:9	**Reserved** Default value=0.
8	**PME_En - RO**. This bit is zero to indicate that PME# assertion from D3 (cold) is disabled. Default value=0.
7:2	**Reserved.** Always returns zero when read, write operations have no effect. Default value=0.
1:0	**PowerState - R/W** This field indicates the current power state of the IGD and can be used to set the IGD into a new power state. If software attempts to write an unsupported state to this field, write operation must complete normally on the bus, but the data is discarded and no state change occurs. On a transition from D3 to D0, the graphics controller is optionally reset to initial values. Bits[1:0]Power State 00 D0 00 D1 10 D2 Not Supported 11 D3 Default value=00.

5.0 Functional Description

5.1 System Address Map

A Mobile Intel® Celeron® processor system based on the Intel® 835 chipset GMCH supports 4 Gbytes of addressable memory space and 64 Kbytes +3 of addressable I/O space. (The P6 bus I/O addressability is 64 Kbytes + 3.) There is a programmable memory address space under the 1 Mbyte region that is divided into regions which can be individually controlled with programmable attributes such as disable, read/write, write-only, or read-only.

The Mobile Intel Celeron processors support addressing of memory ranges larger than 4 Gbytes. The GMCH claims any CPU access over 4 Gbytes and terminates the transaction without forwarding it to hub interface. Simply dropping the data terminates writes and for reads the GMCH returns all zeros on the host bus. Note that the GMCH does not support the PCI Dual Address Cycle (DAC) mechanism and therefore does not allow addressing of greater than 4 Gbytes on the hub interface.

In the following sections, it is assumed that all of the compatibility memory ranges reside on the hub interface/PCI. The exception to this rule is VGA ranges, which may be mapped to IGD. In the absence of more specific references, cycle descriptions referencing PCI should be interpreted as the hub interface/PCI.

5.1.1 System Memory Address Ranges

The Intel 835 chipset GMCH provides a maximum PC133 address decode space of 1.0 Gbyte. The GMCH does not remap APIC memory space. The GMCH does not limit SDRAM space in hardware. **It is the BIOS or system designer's responsibility to limit SDRAM population so that adequate PCI, High BIOS, and APIC memory space can be allocated**. Figure 9 represents the system memory address map in a simplified form and provides additional details on mapping specific memory regions as defined and supported by the Intel 835 chipset.

Figure 9. Memory System Address Map

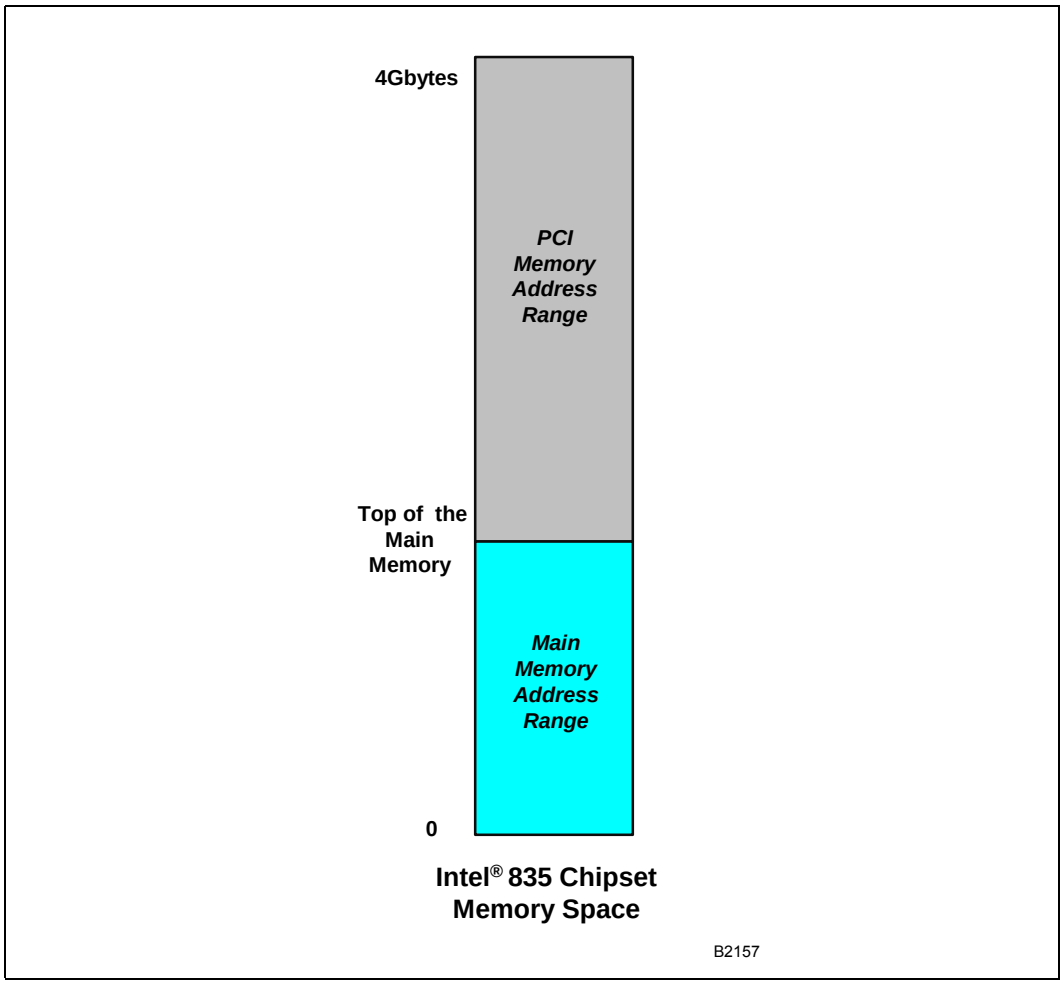
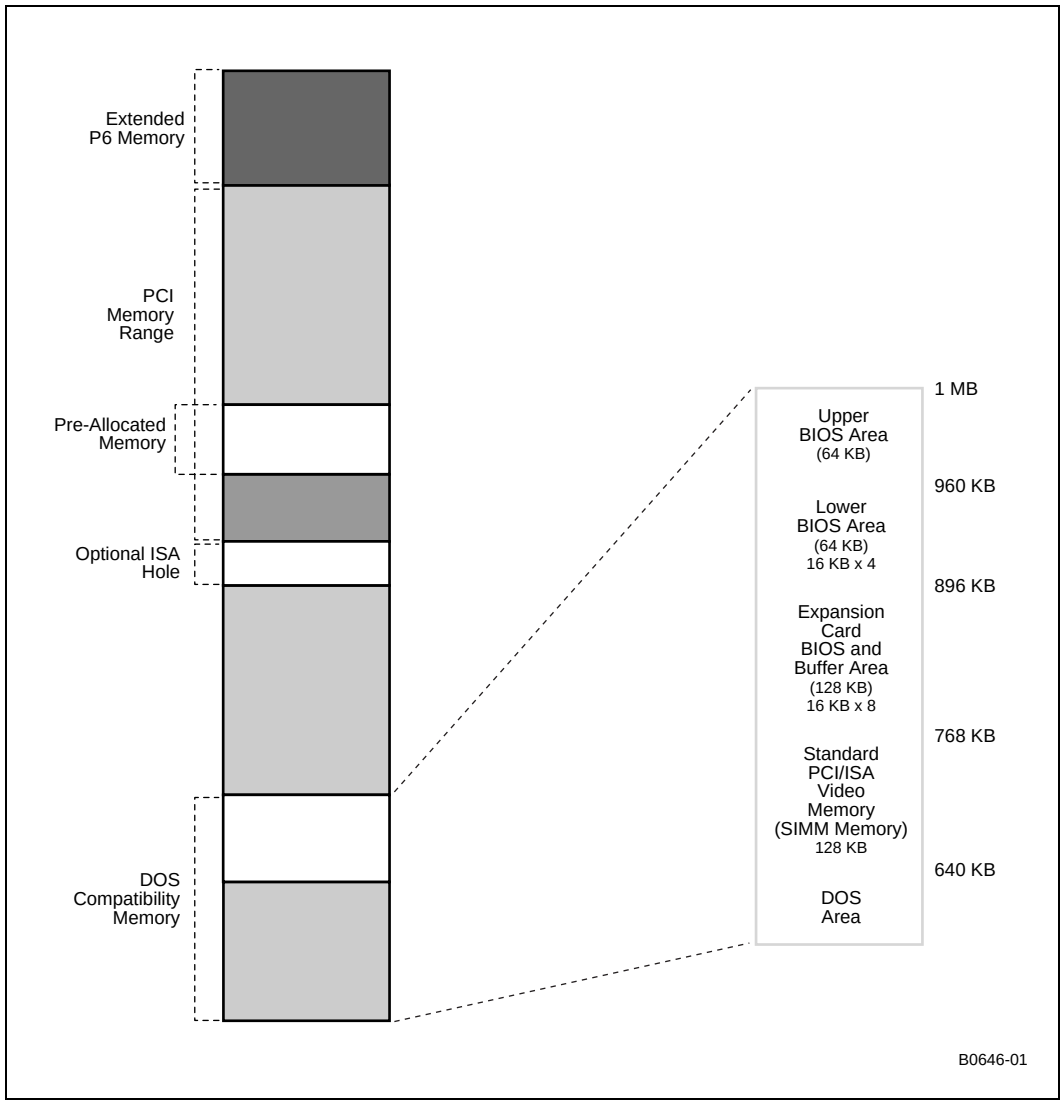


Figure 10. Detailed Memory System Address Map



5.1.2 Compatibility Area

The compatibility area is divided into the following address regions:

- 0 – 640 Kbyte DOS area
- 640 – 768 Kbyte video buffer area
- 768 – 896 Kbytes in 16-Kbyte sections (total of eight sections) - expansion area
- 896 – 960 Kbytes in 16-Kbyte sections (total of four sections) - extended system BIOS area
- 960 Kbytes – 1 Mbyte memory (BIOS area) - system BIOS area

There are 16 memory segments in the compatibility area. Thirteen of the memory ranges can be enabled or disabled independently for both read and write cycles.

Table 81. Memory Segments and Attributes

Memory Segments	Attributes	Comments
000000H - 09FFFFH	Fixed - always mapped to main SDRAM	0 to 640K – DOS Region
0A0000H - 0BFFFFH	Mapped to hub interface, or AGP - configurable as SMM space	Video Buffer (physical SDRAM configurable as SMM space)
0C0000H - 0C3FFFH	WE RE	Add-on BIOS
0C4000H - 0C7FFFH	WE RE	Add-on BIOS
0C8000H - 0CBFFFH	WE RE	Add-on BIOS
0CC000H - 0CFFFFH	WE RE	Add-on BIOS
0D0000H - 0D3FFFH	WE RE	Add-on BIOS
0D4000H - 0D7FFFH	WE RE	Add-on BIOS
0D8000H - 0DBFFFH	WE RE	Add-on BIOS
0DC000H - 0DFFFFH	WE RE	Add-on BIOS
0E0000H - 0E3FFFH	WE RE	BIOS Extension
0E4000H - 0E7FFFH	WE RE	BIOS Extension
0E8000H - 0EBFFFH	WE RE	BIOS Extension
0EC000H - 0EFFFFH	WE RE	BIOS Extension
0F0000H - 0FFFFFH	WE RE	BIOS Area

5.1.2.1 DOS Area (00000h-9FFFFh)

The DOS area is 640 Kbytes in size and always mapped to the main memory controlled by the Intel 835 chipset GMCH.

5.1.2.2 Legacy VGA Ranges (A0000h-BFFFFh)

The legacy 128-Kbyte VGA memory range A0000h-BFFFFh (frame buffer) can be mapped to IGD (device #2) and/or to the hub interface depending on the programming of the VGA steering bits. Priority for VGA mapping is constant in that the Intel 835 chipset GMCH always decodes internally mapped devices first. Internal to the GMCH, decode precedence is always given to IGD. The GMCH always positively decodes an internally mapped device, namely the IGD. Subsequent decoding of regions mapped to the hub interface depends on the legacy VGA configurations bits (VGA enable). This region is also the default for SMM space.

5.1.2.3 Compatible SMRAM Address Range (A0000h-BFFFFh)

When compatible SMM space is enabled, SMM-mode CPU accesses to this range are routed to physical system SDRAM at this address. Non-SMM-mode CPU accesses to this range are considered to be to the video buffer area as described above. Hub interface-originated cycles to enabled SMM space are not allowed and are considered to be to the video buffer area.

5.1.2.4 Expansion Area (C0000h-DFFFFh)

This 128-Kbyte ISA expansion region is divided into eight 16-Kbyte segments. Each segment can be assigned one of four read/write states: read-only, write-only, read/write, or disabled. Typically, these blocks are mapped through GMCH and are subtractively decoded to ISA space. Memory that is disabled is not remapped.

5.1.2.5 Extended System BIOS Area (E0000h-EFFFFh)

This 64-Kbyte area is divided into four 16-Kbyte segments. Each segment can be assigned independent read and write attributes so it can be mapped to either main SDRAM or the hub interface. Typically, this area is used for RAM or ROM. Memory segments that are disabled are not remapped elsewhere.

5.1.2.6 System BIOS Area (F0000h-FFFFFh)

This area is a single 64-Kbyte segment. This segment can be assigned read and write attributes. It is by default (after reset) read/write disabled and cycles are forwarded to hub interface. By manipulating the read/write attributes, the GMCH can “shadow” BIOS into the main SDRAM. When disabled, this segment is not remapped.

5.1.2.7 Extended Memory Area

This memory area covers 100000h (1 Mbyte) to FFFFFFFFh (4 Gbytes-1) address range and it is divided into the following regions:

- Main system SDRAM memory from 1 Mbyte to the top of memory; maximum of 1.0 Gbyte.
- PCI memory space from the top of memory to 4 Gbytes with two specific ranges:
  - — APIC configuration space from FEC0_0000h (4 Gbytes-20 Mbyte) to FECF_FFFFh and FEE0_0000h to FEEF_FFFFh
  - — High BIOS area from 4 Gbytes to 4 Gbytes - 2 Mbytes

5.1.2.8 Main System SDRAM Address Range (0010_0000h to Top of Main Memory)

The address range from 1 Mbyte to the top of main memory is mapped to main SDRAM address range controlled by the Intel 835 chipset GMCH. The Top of Memory (TOM) is limited to 1.0 Gbyte. All accesses to addresses within this range are forwarded by the GMCH to the SDRAM unless a hole in this range is created using the fixed hole as controlled by the FDHC register. Accesses within this hole are forwarded to hub interface.

The GMCH provides a maximum SDRAM address decode space of 4 Gbytes. The GMCH does not remap APIC memory space. The GMCH does not limit SDRAM address space in hardware. It is the BIOS or system designer’s responsibility to limit SDRAM population so that adequate PCI, high BIOS, and APIC memory space can be allocated.

5.1.2.8.1 15 Mbyte-16 Mbyte Window

A hole can be created at 15 Mbyte-16 Mbyte as controlled by the fixed hole enable (FDHC register) in device #0 space. Accesses within this hole are forwarded to the hub interface. The range of physical SDRAM memory disabled by opening the hole is not remapped to the top of the memory – that physical SDRAM space is not accessible. This 15 Mbyte-16 Mbyte hole is an

optionally enabled ISA hole. Video accelerators originally used this hole. Validation and customer SV teams also use it for some of their test cards. That is why it is being supported. There is no inherent BIOS request for the 15-16 hole.

5.1.2.8.2 Pre-allocated Memory

Physical addresses that are not accessible as general system memory and reside within system memory address range (less than TOM) are created for SMM-mode and legacy VGA graphics compatibility. The Intel 835 chipset supports an increased amount of pre-allocated memory to support up to 1600x1200x32 bpp. The pre-allocated memory allows sizes of 512 Kbytes, 1 Mbyte, or 8 Mbytes. The system BIOS must properly initialize these regions.

5.1.2.9 Extended SMRAM Address Range (HSEG and TSEG)

The HSEG and TSEG SMM transaction address spaces reside in this extended memory area.

5.1.2.9.1 HSEG

SMM-mode CPU accesses to enabled HSEG are remapped to 000A0000h-000BFFFFh. Non-SMM-mode CPU accesses to enabled HSEG are considered invalid and are terminated immediately on the host interface. The exceptions to this rule are non-SMM-mode writeback cycles that are remapped to SMM space to maintain cache coherency. Hub interface-originated cycles to enabled SMM space are not allowed. Physical SDRAM behind the HSEG transaction address is not remapped and is not accessible.

5.1.2.9.2 TSEG

TSEG can be up to 1 Mbyte and is at the top of physical memory. SMM-mode CPU accesses to enabled TSEG access the physical SDRAM at the same address. Non-SMM-mode CPU accesses to enabled TSEG is considered invalid and are terminated immediately on the host interface. The exceptions to this rule are non-SMM-mode writeback cycles that are directed to the physical SMM space to maintain cache coherency. Hub interface originated cycles to enabled SMM space are not allowed.

The size of the SMRAM space is determined by the USMM value in the SMRAM register. When the extended SMRAM space is enabled, non-SMM CPU accesses and all other accesses in this range are forwarded to the hub interface. When SMM is enabled, the amount of memory available to the system is equal to the amount of physical SDRAM minus the value in the TSEG register.

5.1.2.10 Intel Dynamic Video Memory Technology (DVMT)

The IGD in the Intel 835 chipset supports DVMT in a non-graphics memory configuration. Intel's dynamic video memory technology is a mechanism that manages system memory and the internal graphics device for optimal graphics performance. DVMT-enabled software drivers, working with the memory arbiter and the operating system, utilize the system memory to support graphics 2D and 3D applications. DVMT dynamically responds to application requirements by allocating the proper amount of display and texturing memory.

5.1.2.11 PCI Memory Address Range (Top of Main Memory to 4 Gbytes)

The address range from the top of main SDRAM to 4 Gbytes (top of physical memory space) supported by the Intel 835 chipset GMCH is normally mapped via the hub interface to PCI.

As an internal graphics configuration, there is one exception to this rule:

Addresses decoded to the memor- mapped range of the internal graphics device. These are forwarded to the internal graphics device.

There are two sub-ranges within the PCI memory address range defined as APIC configuration space and high BIOS address range. As an IGD, the memory-mapped range of the IGD MUST NOT overlap with these two ranges. These ranges are described in detail in the following sections.

5.1.2.12 High BIOS Area (FFE0_0000h -FFFF_FFFFh)

The top 2 Mbytes of the extended memory region is reserved for system BIOS (high BIOS), extended BIOS for PCI devices, and the A20 alias of the system BIOS. CPU begins execution from the high BIOS after reset. This region is mapped to the hub interface so that the upper subset of this region aliases to 16 Mbyte-256 Kbyte range. The actual address space required for the BIOS is less than 2 Mbytes but the minimum CPU MTRR range for this region is 2 Mbytes so that full 2 Mbytes must be considered.

5.2 Host Interface

5.2.1 Overview

The Intel 835 chipset GMCH is optimized for the Mobile Intel Celeron processors. The GMCH supports a PSB frequency of 133 MHz using 1.25 V AGTL signaling. The AGTL buffers support dual-ended termination. The GMCH supports 32-bit host addressing, decoding up to 4 Gbytes of memory address space for the processor. CPU memory writes to address space above 4 Gbytes are immediately terminated and discarded. CPU memory reads to address space above 4 Gbytes are immediately terminated and return the value of the pulled-up GTL host bus. Host initiated I/O cycles are decoded to PCI1, hub interface, or GMCH configuration space. Host-initiated memory cycles are decoded to PCI1, hub interface, or system SDRAM, or graphics memory mapped registers. Host cycles to the integrated graphics device, or hub interface, are subject to dynamic deferring.

5.2.2 Processor-Unique PSB Activity

The Intel 835 chipset GMCH recognizes and supports a large subset of the transaction types that are defined for the P6 bus interface. However, each of these transaction types has a multitude of response types, some of which are not supported by this controller. All transactions are processed in the order that they are received on the host bus. A summary of transactions supported by the GMCH is given in Table 82.

Table 82. Host Bus Transactions Supported by GMCH (Sheet 1 of 2)

Transaction	REQa[4:0]#	REQb[4:0]#	GMCH Support
Deferred Reply	0 0 0 0 0	X X X X X	The GMCH initiates a deferred reply for a previously deferred transaction.
Reserved	0 0 0 0 1	X X X X X	Reserved
Interrupt Acknowledge	0 1 0 0 0	0 0 0 0 0	Interrupt acknowledge cycles are forwarded to the hub interface bus.
Special Transactions	0 1 0 0 0	0 0 0 0 1	See Section 5.2.4.14.

Table 82. Host Bus Transactions Supported by GMCH (Sheet 2 of 2)

Transaction	REQa[4:0]#	REQb[4:0]#	GMCH Support
Reserved	0 1 0 0 0	0 0 0 1 x	Reserved
Reserved	0 1 0 0 0	0 0 1 x x	Reserved
Branch Trace Message	0 1 0 0 1	0 0 0 0 0	The GMCH terminates a branch trace message without latching data.
Reserved	0 1 0 0 1	0 0 0 0 1	Reserved
Reserved	0 1 0 0 1	0 0 0 1 x	Reserved
Reserved	0 1 0 0 1	0 0 1 x x	Reserved
I/O Read	1 0 0 0 0	0 0 x LEN#	I/O read cycles are forwarded to hub interface unless they target the GMCH configuration space (this includes the IGD). In this case, the GMCH picks up the transaction.
I/O Write	1 0 0 0 1	0 0 x LEN#	I/O write cycles are forwarded to hub unless they target the GMCH configuration space (this includes the IGD). In this case, the GMCH picks up the transaction.
Reserved	1 1 0 0 x	0 0 x x x	Reserved
Memory Read & Invalidate	0 0 0 1 0	0 0 x LEN#	Host-initiated memory read and invalidate cycles are forwarded to system SDRAM, hub interface, or graphics memory mapped registers. The GMCH initiates an MRI (LEN=0) cycle to snoop a hub interface, or cacheable IGD-initiated write cycle to system SDRAM.
Reserved	0 0 0 1 1	0 0 x LEN#	Reserved
Memory Code Read	0 0 1 0 0	0 0 x LEN#	Memory code read cycles are forwarded to system SDRAM, or hub interface.
Memory Data Read	0 0 1 1 0	0 0 x LEN#	Host-initiated memory read cycles are forwarded to system SDRAM, hub interface, or graphics memory mapped registers. The GMCH initiates a memory read cycle to snoop a hub interface, or cacheable IGD-initiated read cycle to system SDRAM.
Memory Write (no retry)	0 0 1 0 1	0 0 x LEN#	This memory write is a writeback cycle and cannot be retried. The GMCH forwards the write to system SDRAM.
Memory Write (can be retried)	0 0 1 1 1	0 0 x LEN#	The memory write cycle is forwarded to system SDRAM, hub interface, or graphics memory mapped registers.

For Memory cycles, REQa[4:3]# = ASZ#. The GMCH only supports ASZ# = 00 (32 bit address).

REQb[4:3]# = DSZ#. For the Intel Pentium Pro processor, DSZ# = 00 (64 bit data bus size).

LEN# = data transfer length as follows:

LEN#	Data length
00	<= 8 bytes (BE[7:0]# specify granularity)
01	Length = 16 bytes BE[7:0]# all active
10	Length = 32 bytes BE[7:0]# all active
11	Reserved.

Table 83. Host Bus Responses Supported by GMCH

RS2#	RS1#	RS0#	Description	GMCH Support
0	0	0	Idle	
0	0	1	Retry Response	This response is generated if an access is directed to a resource that cannot be accessed by the processor at this time and the logic must avoid deadlock. Hub interface-directed reads and writes, SDRAM locked reads, and IGD reads and writes can be retried. Unless there is an attempt to establish LOCK, the GMCH never retries a cycle that targets system memory.
0	1	0	Deferred Response	This response can be returned for all transactions that can be executed 'out of order.' Hub interface-directed reads (memory, I/O and interrupt acknowledge) and writes (I/O only), and IGD directed reads (memory and I/O) and writes (I/O only) can be deferred. Unless there is an attempt to establish LOCK, the GMCH never defers a cycle that targets system memory.
0	1	1	Reserved	Reserved
1	0	0	Hard Failure	Not supported
1	0	1	No Data Response	This is for transactions where the data has been already transferred or for transactions where no data is transferred. Writes and zero-length reads receive this response.
1	1	0	Implicit Writeback	This response is given for those transactions where the initial transactions snoop hits on a modified cache line.
1	1	1	Normal Data Response	This response is for transactions where data accompanies the response phase. Reads receive this response.

5.2.3 Host Addresses Above 4 Gbytes

CPU memory writes to address space above 4 Gbytes is terminated and discarded immediately. CPU memory reads to address space above 4 Gbytes also are immediately terminated and return the value of the pulled-up GTL host bus.

5.2.4 Host Bus Cycles

The following transaction descriptions illustrate the various operations in their most straightforward representation. The diagrams do not attempt to show the transaction phase relationships when multiple transactions are active on the CPU bus. For a full description of the CPU Bus functionality please refer to the latest *P6 Family of Processor Hardware Developer's Manual.*

5.2.4.1 Partial Reads

Partial read transactions include: I/O reads and memory read operations less than or equal to 8 bytes (four consecutive bytes for I/O) within an aligned 8-byte span. The byte enable signals, BE#[7:0], select which bytes in the span to read.

5.2.4.2 Part-Line Read and Write Transactions

The Intel 835 chipset GMCH does not support a part-line, i.e., 16-byte transactions.

5.2.4.3 Cache Line Reads

A read of a full cache line (as indicated by the LEN[1:0]=10 during request phase) requires 32 bytes of data to be transferred, which translates into four data transfers for a given request. If selected as a target, the Intel 835 chipset GMCH determines if the address is directed to system SDRAM, graphics SDRAM, or hub interface, and provides the corresponding command and control to complete the transaction.

5.2.4.4 Partial Writes

Partial write transactions include: I/O and memory write operations of 8 bytes or less (maximum of 4 bytes for I/O) within an aligned 8-byte span. The byte enable signals, BE#[7:0], select which bytes in the span to write. I/O writes crossing a 4-byte boundary are broken into two separate transactions by the CPU.

5.2.4.5 Cache Line Writes

A write of a full cache line requires 32 bytes of data to be transferred, which translates into four data transfers for a given request.

5.2.4.6 Memory Read and Invalidate (Length > 0)

A Memory Read and Invalidate (MRI) transaction is functionally equivalent to a cache line read. The purpose this special transaction is to support write allocation (write miss case) of cache lines in the processors. When a processor issues an MRI, the cache line is read as in a normal cache line read operation; however, all other caching agents must invalidate this line if they have it in a shared or exclusive state. If a caching agent has this line in the modified state, then it must be written back to memory and invalidated. The Intel 835 chipset GMCH snarfs the writeback data.

5.2.4.7 Memory Read and Invalidate (Length = 0)

A memory read and invalidate transaction of length zero, MRI(0) does not have an associated data response. Executing the transaction informs other agents in the system that the agent issuing this request wants exclusive ownership of a cache line that is in the shared state (write hit to a shared line). Agents with this cache line invalidate the line. If this line is in the modified state an implicit writeback cycle is generated and the Intel 835 chipset GMCH snarfs the data.

The Intel 835 chipset GMCH generates length=0 memory read and invalidates transactions for hub interface, or IGD memory write cycles to system SDRAM.

5.2.4.8 Memory Read (Length = 0)

A memory read of length zero, MR(0), does not have an associated data response. This transaction is used by the GMCH to snoop for the hub interface to system SDRAM-snoopable system SDRAM read accesses, and IGD-snoopable system SDRAM read accesses. The Intel 835 chipset GMCH snoop request policy is identical for hub interface and IGD memory read transactions.

Note that the GMCH performs single MR(0) cycles for hub interface reads less than or equal to 32 bytes, for read lines directed to system SDRAM, and for IGD-cacheable reads or read lines (which can only be directed to system SDRAM). The GMCH does multiple snoop-ahead cycles for hub interface burst reads greater than 32. Multiple snoop-ahead cycles by the GMCH are not necessary for the IGD as burst reads are not supported by the IGD.

5.2.4.9 Host-Initiated Zero-Length R/W Cycles

Streaming SIMD Extension (SSE) new instructions can result in zero-length read and write cycles to the chipset.

The Intel 835 chipset family GMCH supports a zero-length processor write cycle by executing a 1 Qword write cycle to the targeted destination with all 8 byte enables turned off. The following destinations for host initiated zero-length writes are supported:

- Coherent system memory
- Aperture mapped to system memory
- Aperture mapped to graphics memory
- GMCH internal memory-mapped I/O registers
- PCI (via hub interface)

The GMCH only supports zero-length processor read cycles that target coherent system memory. When targeting coherent system memory, the GMCH forwards the cycle as a 1 Qword read from system SDRAM. The data is returned to the GMCH. The GMCH then returns a “no data” response to the host and empties the returned data from its buffer.

5.2.4.10 Cache Coherency Cycles

The Intel 835 chipset GMCH generates an implicit writeback response during host bus read and write transactions when a CPU asserts H_HITMB during the snoop phase. The CPU-initiated write case has two data transfers, the requesting agents data followed by the snooping agent’s writeback data.

The GMCH performs a memory read and invalidate cycle of length = 0 (MRI[0]) on the CPU bus when a hub interface, or IGD snoopable system SDRAM write cycle occurs.

The GMCH performs a memory read cycle with length = 0 (MR[0]) on the CPU bus when a hub interface, or IGD-snoopable system SDRAM read cycle occurs.

5.2.4.11 Interrupt Acknowledge Cycles

A processor agent issues an interrupt acknowledge cycle in response to an interrupt from an 8259-compatible interrupt controller. The interrupt acknowledge cycle is similar to a partial read transaction, except that the address bus does not contain a valid address. An interrupt acknowledge cycle is always directed to the hub interface and never to the IGD.

5.2.4.12 Locked Cycles

The Intel 835 chipset GMCH supports resource locking due to the assertion of the H_LOCKB line on the CPU bus as follows.

5.2.4.12.1 CPU – System SDRAM Locked Cycles

The Intel 835 chipset GMCH supports CPU to SDRAM locked cycles. The host bus may not execute any other transactions until the locked cycle is complete. The GMCH arbiter may grant another hub interface, but any “coherent” cycles to SDRAM are blocked. CPU lock operations DO NOT block any “non-coherent” accesses to SDRAM.

5.2.4.12.2 CPU – Hub Interface Locked Cycles

Any CPU-to-hub interface locked transaction initiates a hub interface locked sequence. The P6 bus implements the bus lock mechanism, which means that no change of bus ownership can occur from the time one agent has established a locked transaction (i.e., the initial read cycle of a locked transaction has completed) until the locked transaction is completed. Note that for CPU-to-hub interface lock transactions, a bit in the request packet indicates a lock transaction.

Any concurrent cycle that requires snooping on the host bus is not processed while a LOCK transaction is occurring on the host bus.

Hub interface-to-SDRAM locked cycles are not supported.

5.2.4.12.3 CPU – IGD (Graphics Memory)

The IGD does not support locked operations and therefore both CPU-locked and non-locked transactions destined to IGD graphics memory are propagated in the same manner. Note however, that any concurrent cycle that requires snooping on the host bus is not processed while a LOCK transaction occurs on the host bus.

5.2.4.13 Branch Trace Cycles

An agent issues a branch trace cycle for taken branches if execution tracing is enabled. Address Aa [35:3]# is reserved and can be driven to any value. D [63:32]# carries the linear address of the instruction causing the branch and D [31:0]# carries the target linear address. The GMCH responds and retires this transaction but does not latch the value on the data lines or provide any additional support for this type of cycle.

5.2.4.14 Special Cycles

A special cycle is defined when REQa[4:0] = 01000 and REQb[4:0]= xx001. In the first address phase Aa [35:3]# is undefined and can be driven to any value. In the second address phase, Ab [15:8]# defines the type of special cycle issued by the processor. All host-initiated special cycles are routed to hub interface.

Special cycles are “posted” into the Intel 835 chipset GMCH. The host bus transaction is terminated immediately. It does not wait for the cycle to propagate or terminate on hub interface.

Table 84 specifies the cycle type and definition as well as the action taken by the GMCH when the corresponding cycles are identified.

Table 84. Intel® 835 Chipset GMCH Responses to Host Initiated Special Cycles

BE[7:0}#	Special Cycle Type	Action Taken
0000 0000	NOP	This transaction has no side effects.
0000 0001	Shutdown	This transaction is issued when an agent detects a severe software error that prevents further processing. This cycle is claimed by the GMCH and propagated as a shutdown special cycle over the hub interface bus. This cycle is retired on the CPU bus after the associated hub interface special cycle request packet is successfully broadcast over hub interface.
0000 0010	Flush	This transaction is issued when an agent has invalidated its internal caches without writing back any modified lines. The GMCH claims this cycle and simply retires it.
0000 0011	Halt	This transaction is issued when an agent executes a HLT instruction and stops program execution. This cycle is claimed by the GMCH and propagated over the hub interface as a halt special cycle. This cycle is retired on the CPU bus after the associated hub interface special cycle request packet is successfully broadcast over the hub interface.
0000 0100	Sync	This transaction is issued when an agent has written back all modified lines and has invalidated its internal caches. The GMCH claims this cycle and simply retires it.
0000 0101	Flush Acknowledge	This transaction is issued when an agent has completed a cache sync-and-flush operation in response to an earlier FLUSH# signal assertion. The GMCH claims this cycle and retires it.
0000 0110	Stop Clock Acknowledge	This transaction is issued when an agent enters stop clock mode. This cycle is claimed by the GMCH and propagated over hub interface as a stop grant special cycle. This cycle is retired on the CPU bus after the associated hub interface special cycle request packet is successfully broadcast over hub interface.
0000 0111	SMI Acknowledge	This transaction is first issued when an agent enters the system management mode (SMM). Ab[7]# is also set at this entry point. All subsequent transactions from the CPU with Ab [7]# set are treated by the GMCH as accesses to the SMM space. No corresponding cycle is propagated to the hub interface. To exit the system management mode the CPU issues another one of these cycles with the Ab [7]# bit deasserted. The SMM space access is closed by the GMCH at this point.
All others	Reserved	

5.2.5 In-Order Queue Pipelining

All agents on the CPU bus track the number of pipelined bus transaction with an In-Order Queue (IOQ). The GMCH can support an IOQ depth of eight and uses BNR# to guarantee that limit is not exceeded.

5.2.6 Write Combining

To allow for high speed write capability for graphics, the Uncacheable, Speculative, Write-Combining (USWC) memory type provides a write-combining buffering mechanism for write operations. A high percentage of graphics transactions are writes to the memory-mapped graphics region, normally known as the linear frame buffer. Reads and writes to USWC are non-cached and can have no side effects.

In the case of graphics, current 32-bit drivers (without modifications) would use partial write protocol to update the frame buffer. The highest performance write transaction on the CPU bus is the line write. By combining several back-to-back partial write transactions (internal to the CPU) into a line write transaction on the CPU bus, the performance of frame buffer accesses would be greatly improved. To this end, the CPU supports the USWC memory. Writes to USWC memory can be buffered and combined in the processor's write-combining buffers (WCB). The WCB is flushed after executing a serializing locked, I/O instruction, or the WCB is full (32 bytes). The WCB can be flushed under different situations (see ***Note*** below). In order to extend this capability to the current drivers, it is necessary to set up the linear frame buffer address range to be USWC memory type. Programming the MTRR registers in the CPU can do this.

If the number of bytes in the WCB is less than 32 then a series of write operations less than or equal to 8 bytes are performed upon WCB flushing. The GMCH further optimizes the procedure by providing write combining for CPU-to-hub interface, and CPU-to-IGD write transactions. If the target of CPU writes is hub interface memory, then the data is combined and sent to the hub interface bus as a single write burst. The same concept applies to CPU writes to IGD memory. The USWC writes that target system SDRAM are handled as regular system SDRAM writes.

Note that the application of USWC memory attribute is not limited only to the frame buffer support and that the GMCH-M implements write combining for any CPU-to-hub interface posted write.

Note: Refer to the following documents on how to implement write combining buffers:

- *Intel® Write Combining Memory Implementation Guidelines* (24422) http://developer.intel.com/design/PentiumII/applnots/244422.htm
- *Intel® Architecture Software Developer's Manual Volume 3 System Programming Guide* (245572) http://developer.intel.com/design/Pentium4/manuals/245472.htm

5.3 System Memory Interface

5.3.1 SDRAM Interface Overview

The Intel 835 chipset GMCH integrates a main memory SDRAM controller with a 64-bit wide interface. The GMCH memory buffers support LVTTL (SDRAM) signaling at 133 MHz.

Configured for single data rate SDRAM, the GMCH memory interface includes support for:

- Up to 512 Mbytes of 133-MHz SDRAM using 256-Mbit technology
- PC133 DIMMs
- Maximum of twoDIMMs, single-sided and/or double-sided
- Four bank memory technologies
- Four integrated clock buffers

The 2-bank select lines SMBA [1:0] and the 13 address lines SMMA [12:0] allow the Intel 835 chipset to support 64-bit wide DIMMs using 64-Mbit, 128-Mbit, and 256-Mbit SDRAM technology. While address lines SMMA [9:0] determine the starting address for a burst, burst lengths are fixed at four. Six chip selects SMCSB lines allow maximum of three rows of single-sided DIMMs and six rows of double-sided SDRAM DIMMs.

The Intel 835 chipset GMCH main memory controller targets CAS latencies of two and three for SDRAM. The chipset provides refresh functionality with programmable rate (normal SDRAM rate is one refresh/15.6 ms). For write operations of less than a Qword in size, a byte-wise write is performed.

5.3.2 SDRAM Organization and Configuration

In the following discussion the term “row” refers to a set of memory devices that are simultaneously selected by a SMCSB signal. The Intel 835 chipset supports a maximum of 4 rows of memory. For the purposes of this discussion, a “side” of a DIMM is equivalent to a “row” of SDRAM devices.

The 2-bank select lines SMBA [1:0] and the 13 address lines SMMA [12:0] allow the Intel 835 chipset to support 64-bit wide DIMMs using x16 64-Mbit, 128-Mbit, and 256-Mbit SDRAM technologies.

Table 85. System Memory DIMM Configurations

SDRAM Technology (Density)	Dev Depth	Dev Width	Dev Per Side	Capacity Per Side	# Of Row Addr Bits	# Of Col Addr Bits	# Of Bank Addr Bits	Page Size	Max Capacity SDR (Two DIMMs)
64 Mbits	4 Mbits	X16	4	32 Mbytes	12	8	2	2 Kbytes	128 Mbytes
128 Mbits	8 Mbits	X16	4	64 Mbytes	12	9	2	4 Kbytes	256 Mbytes
256 Mbits	16 Mbits	X16	4	128 Mbytes	13	9	2	4 Kbytes	512 Mbytes

5.3.2.1 Configuration Mechanism for DIMMs

Detection of the type of SDRAM installed on the DIMM is supported using a serial presence detect mechanism as defined in the JEDEC DIMM specification. This uses the SCL, SDA and SA[2:0] pins on the DIMMs to detect the type and size of the installed DIMMs. No special programmable modes are provided on the Intel 835 chipset for detecting the size and type of memory installed. Type and size detection must be done using the serial presence detection pins.

5.3.2.1.1 Memory Detection and Initialization

Before any cycles to the memory interface can be supported, the Intel 835 chipset SDRAM registers must be initialized. The Intel 835 chipset must be configured for operation with the installed memory types. Detection of memory type and size is done through the System Management Bus (SMBus) interface on the ICH4. This 2-wire bus is used to extract the SDRAM type and size information from the serial presence detect port on the SDRAM DIMMs. SDRAM DIMMs contain a 5-pin serial presence detect interface, including SCL (serial clock), SDA (serial data) and SA[2:0]. Devices on the SMBus have a 7-bit address. For the SDRAM DIMMs, the

upper 4 bits are fixed at 1010. The lower three bits are strapped on the SA[2:0] pins. SCL and SDA are connected directly to the system management bus on the ICH4. Data is read from the serial presence detect port on the DIMMs using a series of I/O cycles to the south bridge. BIOS needs to determine the size and type of memory used for each of the rows of memory to properly configure the Intel 835 chipset memory interface.

5.3.2.1.2 SDRAM Register Programming

This section provides an overview of how the required information for programming the SDRAM registers is obtained from the serial presence detect ports on the DIMMs. The serial presence detect ports are used to determine refresh rate, MA and MD buffer strength, row type (on a row-by-row basis), SDRAM timings, row sizes, and row page sizes. Table 86 lists a subset of the data available through the on-board serial presence detect ROM on each DIMM.

Table 86. Data Bytes on DIMM Used for Programming SDRAM Registers

Byte	Function
2	Memory type (EDO, SDR SDRAM)
3	# of Row addresses, not counting bank addresses
4	# of Column addresses
5	# of Banks of SDRAM (Single or Double sided DIMM)
11	ECC, no ECC
12	Refresh rate
17	# of Banks on each device
36-41	Access time from clock for CAS# latency 1 through 7
42	Data width of SDRAM components
126	Memory frequency

Table 86 is only a subset of the defined SPD bytes on the DIMMs. These bytes collectively provide enough data for programming the Intel 835 chipset SDRAM registers.

5.3.3 SDRAM Address Translation and Decoding

The Intel 835 chipset contains address decoders that translate the address received on the host bus, or the hub interface to an effective memory address. Decoding and translation of these addresses vary with the three SDRAM types. The number of pages, page sizes, and densities supported vary with the 4 SDRAM types. In general, the Intel 835 chipset supports 64-Mbit, 128-Mbit, and 256-Mbit SDRAM devices. The multiplexed row/column address to the SDRAM memory array is provided by the SMBA [1:0] and SMMA [12:0] signals. These addresses are derived from the host address bus as defined in Table 86 for SDRAM devices.

Table 87. Address Translation and Decoding

Tech	Depth	Width	Address Usage			Row	Page	BS	BS	MA	MA	MA	MA	MA	MA	MA	MA	MA	MA	MA	MA
			Row	Col	Bank	Size		1	0	12	11	10	9	8	7	6	5	4	3	2	1	0
64 Mbits	4 Mbits	16	12	8	2	32 Mbytes	2K	12	11	X	15	14	13	24	23	22	21	20	19	18	17	16
								12	11	X	X	PA	X	X	10	9	8	7	6	5	4	3
128 Mbits	8 Mbits	16	12	9	2	64 Mbytes	4K	13	12	X	15	14	25	24	23	22	21	20	19	18	17	16
								13	12	X	X	PA	X	11	10	9	8	7	6	5	4	3
256 Mbits	16 Mbits	16	13	9	2	128 Mbytes	4K	13	12	15	14	26	25	24	23	22	21	20	19	18	17	16
								13	12	X	X	PA	X	11	10	9	8	7	6	5	4	3

5.3.4 SDRAM Performance Description

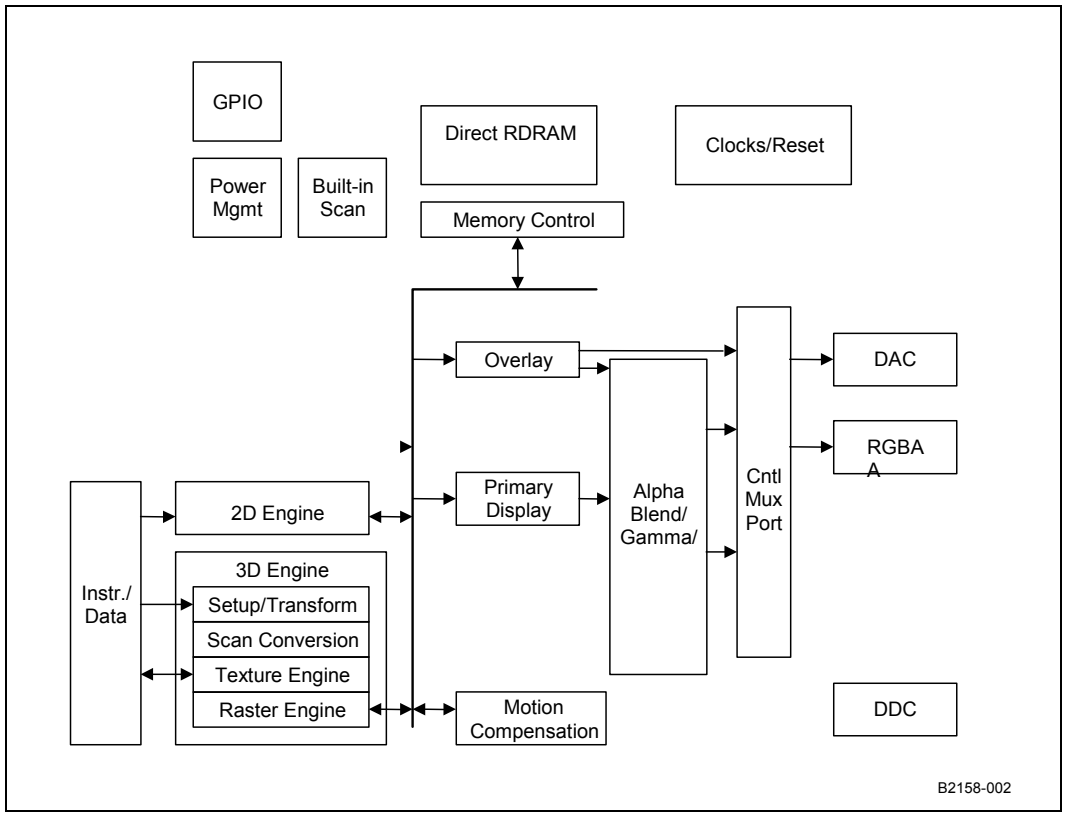
The overall SDRAM performance is controlled by the SDRAM timing register, pipelining depth used in the Intel 835 chipset, SDRAM speed grade, and the type of SDRAM used in the system. Besides this, the exact performance in a system is dependent on the total memory supported, external buffering and memory array layout. The most important contribution to overall performance by the system memory controller is to minimize the latency required to initiate and complete requests to memory, and to support the highest possible bandwidth (full streaming, quick turn-arounds). One measure of performance is the total flight time to complete a cache line request. A true discussion of performance involves the entire chipset, not just the system memory controller.

5.4 Internal Graphics Description

The Intel 835 chipset GMCH provides a highly integrated graphics accelerator and PCI set while allowing a flexible integrated system graphics solution.

The Intel 835 GMCH is not legacy VGA compatible. The device does not support VGA/EGA display modes.

Figure 11. Intel® 835 Chipset GMCH Graphics Block Diagram



High bandwidth access to data is provided through the system memory port. The Intel 835 chipset GMCH can access UMA memory located in system memory at 1.06 Gbytes/s. The Intel 835 chipset uses a tiling architecture to minimize page miss latencies and maximize effective rendering bandwidth.

5.4.1 3D/2D Instruction Processing

The Intel 835 chipset GMCH contains an extensive set of instructions that control various functions including 3D rendering, BLT operations, display, MPEG decode acceleration, and overlay. The 3D instructions set 3D pipeline states and control the processing functions. The 2D instructions provide an efficient method for invoking BLT operations.

5.4.2 3D Engine

The 3D engine of the Intel 835 chipset GMCH has been designed with a deep-pipelined architecture, where performance is maximized by allowing each stage of the pipeline to simultaneously operate on different primitives or portions of the same primitive. The GMCH supports perspective-correct texture mapping, multi textures, bump-mapping, cubic environment maps, bilinear, trilinear and anisotropic MIP mapped filtering, Gouraud shading, alpha-blending, vertex- and per-pixel fog, and Z/W buffering. These features are independently enabled (disabled) via a set of 3D instructions.

The 3D pipeline subsystem performs the 3D rendering acceleration. The main blocks of the pipeline are the setup engine, scan converter, texture pipeline, and raster pipeline. A typical programming sequence would send instructions to set the state of the pipeline followed by rendering instructions containing 3D primitive vertex data.

5.4.2.1 Set-up Engine

The Intel 835 chipset GMCH 3D set-up engine takes the input data associated with each vertex of a 3D primitive and computes the various parameters required for scan conversion. In formatting this data, GMCH maintains sub-pixel accuracy. The per-vertex data is converted io gradients that can be used to interpolate the data at any pixel within a polygon (colors, alpha, Z depth, fog, and texture coordinates). The pixels covered by a polygon are identified and per-pixel texture addresses are calculated.

5.4.2.2 Viewport Transform and Perspective Divide

The 3D-geometry pipeline involves transformation of vertices from model space to clipping space followed by clip test and clipping. Lighting can be performed during the transformation or at any other point in the pipeline. After clipping, the next stage is perspective divide followed by transformation to the viewport or screen space. The Intel 835 chipset GMCH supports viewport transform and perspective-divide portion of the 3D geometry pipeline in hardware.

5.4.2.3 3D Primitives and Data Formats Support

The 3D primitives rendered by the Intel 835 chipset GMCH are points, lines, discrete triangles, line strips, triangle strips, triangle fans, and polygons. In addition, GMCH supports Flexible Vertex Format (FVF) from Microsoft's DirectX* 6.0 which enables the application to specify a variable length of parameter list obviating the need to send unused information to the hardware. Strips, fans, and indexed vertices as well as FVF improves the vertex rate delivered to the set-up engine significantly.

5.4.2.4 Pixel-Accurate Fast Scissoring and Clipping Operation

The Intel 835 chipset GMCH supports clipping to a scissoring rectangle within the drawing window. GMCH's clipping and scissoring in hardware reduce the need for software to process polygons and improves performance. During the set-up stage, GMCH clips polygons to the drawing window. The scissor rectangle accelerates the clipping process by allowing the driver to clip to a bigger region than the hardware renders. The scissor rectangle needs to be pixel-accurate, and independent of line and point width. GMCH supports a single scissor-box rectangle.

5.4.2.5 Backface Culling

As part of the setup, the Intel 835 chipset GMCH discards polygons from further processing if they are facing away from or towards the user's viewpoint. This operation, referred to as "backface culling" is accomplished based on the "clockwise" or "counter-clockwise" orientation of the vertices on a primitive. This can be enabled or disabled by the driver.

5.4.2.6 Scan Converter

The scan converter takes the vertex and edge information used to identify all pixels that are affected by features being rendered. It works on a per-polygon basis, and one polygon may be entering the pipeline while calculations finish on another.

5.4.2.7 Texture Engine

As texture sizes increase beyond the bounds of graphics memory, it becomes impractical to execute textures from graphics memory. The Intel 835 chipset GMCH, using Intel's direct memory execution model, simplifies this process by rendering each scene using the texture located in system memory. The 835 chipset includes a cache controller to avoid frequent memory fetches of recently used texture data.

The Intel 835 chipset allows an image, pattern, or video to be placed on the surface of a 3D polygon. The texture engine performs texture color or chromakey matching, texture filtering (anisotropic, trilinear, and bilinear interpolation), and YUV to RGB conversions.

5.4.2.8 Perspective Correct Texture Support

Mapping a 2D texture pattern onto each pixel of the polygon generates a textured polygon. A texture map is like wallpaper pasted onto the polygon. Because polygons are rendered in perspective, it is important that texture be mapped in perspective as well. Without perspective correction, texture is distorted when an object recedes into the distance. Perspective correction performs a computation-intensive "per-pixel-divide" operation on each pixel. Perspective correction is necessary for realistic 3D graphics.

5.4.2.9 Texture Decompression

Microsoft's DirectX* 7.0 supports texture compression to reduce the bandwidth required to deliver textures. As the textures' average sizes (512x512) get larger with higher color depth and multiple textures become the norm, it becomes increasingly important to provide a mechanism to compress textures. The Intel 835 chipset GMCH supports DX7 decompression. Texture decompression formats supported include DXT1, DXT2, DXT3, DXT4, and DXT5.

5.4.2.10 Texture ColorKey and ChromaKey

ColorKey and ChromaKey describe two methods of removing a specific color or range of colors from a texture map before it is applied to an object. For "nearest" texture filter modes, removing a color simply makes those portions of the object transparent (the previous contents of the back buffer show through). For "linear" texture filtering modes, the texture filter is modified if only the non-nearest neighbor texture map elements or "texels" match the key (range).

ColorKeying occurs with paletted textures and removes colors according to an index (before the palette is accessed). When a color palette is used with indices to indicate a color in the palette, the indices can be compared to a state variable "ColorKey Index Value" and if a match occurs and ColorKey is enabled, then this value's contribution is removed from the resulting pixel color. The Intel 835 chipset GMCH defines index matching as ColorKey.

ChromaKeying can be performed for both paletted and non-paletted textures, and removes texels that fall within a specified color range. The ChromaKey mode refers to testing the ARGB or YUV components to see if they fall between high and low state variable values. If the color of a texel contribution is in this range and ChromaKey is enabled, then this contribution is removed from the resulting pixel color.

5.4.2.11 Anti-aliasing

Aliasing is one of the artifacts that degrades image quality. In its simplest manifestation, aliasing causes the jagged staircase effects on sloped lines and polygon edges. Another artifact is the moiré patterns that occur as a result of a very small number of pixels available on screen to contain the data of a high-resolution texture map.

Full scene anti-aliasing uses supersampling, so that the image is rendered internally at a higher resolution than is displayed on screen. The Intel 835 chipset GMCH can render internally at 1600x1200 and then this image is down-sampled (via a bilinear filter) to the screen resolution of 640x480 and 800x600. Full scene anti-aliasing removes “jaggies” at the edges as well as moiré patterns. The GMCH renders the super-sampled image up to 2Kx2K pixel dimensions. The GMCH then reads it as a texture and bilinear filters it to the final resolution.

5.4.2.12 Texture Map Filtering

Many texture-mapping modes are supported. Perspective-correct mapping is always performed. As the map is fitted across the polygon, the map can be tiled, mirrored in either the U or V directions, or mapped up to the end of the texture and no longer placed on the object (this is known as clamp mode). The way a texture is combined with other object attributes is also definable.

The Intel 835 chipset GMCH supports up to 12 Levels-of-Detail (LODs) ranging in size from 2048x2048 to 1x1 texels. Included in the texture processor is a texture cache, which provides efficient MIP-mapping.

The Intel 835 chipset GMCH supports seven types of texture filtering:

- Nearest (also known as point filtering): Texel with coordinates nearest to the desired pixel is used. (This is used if only one LOD is present).
- Linear (also known as bilinear filtering): A weighted average of a 2x2 area of texels surrounding the desired pixel are used. (This is used if only one LOD is present).
- Nearest MIP Nearest (also known as point filtering): This is used if many LODs are present. The nearest LOD is chosen and the texel with coordinates nearest to the desired pixel are used.
- Linear MIP Nearest (bilinear MIP mapping): This is used if many LODs are present. The nearest LOD is chosen and a weighted average of a 2x2 area of texels surrounding the desired pixel is used (four texels). This is also referred to as bilinear MIP mapping.
- Nearest MIP Linear (Point MIP Mapping): This is used if many LODs are present. Two appropriate LODs are selected and within each LOD the texel with coordinates nearest to the desired pixel are selected. The final texture value is generated by linear interpolation between the two texels selected from each of the MIP maps.
- Linear MIP Linear (Trilinear MIP Mapping): This is used if many LODs are present. Two appropriate LODs are selected and a weighted average of a 2x2 area of texels surrounding the desired pixel in each MIP map is generated (four texels per MIP map). The final texture value is generated by linear interpolation between the two texels generated for each of the MIP maps. Trilinear MIP mapping is used to minimize the visibility of LOD transitions across the polygon.
- Anisotropic MIP Nearest (Anisotropic Filtering): This is used if many LODs are present. The nearest LOD-1 level is determined for each of four sub-samples for the desired pixel. These four sub-samples are filtered bilinearly and averaged together.

Both Microsoft’s DirectX* 6.0 and Open GL* (rev.1.1) allow support for all these filtering modes.

5.4.2.13 Multiple Texture Composition

The Intel 835 chipset GMCH also performs multiple texture composition. This allows the combination of two or more MIP maps to produce a new one with new LODs and texture attributes in a single or iterated pass. The set-up engine supports up to four texture map coordinates in as single pass. GMCH allows up to two bilinear MIP maps or a single trilinear MIP map to be composited in a single pass. More than two bilinear MIP maps or more than one trilinear MIP map would require multiple passes. The actual blending or composition of the MIP maps is done in the raster engine. The texture engine provides the required texels including blending information.

Flexible vertex format support allows multi-texturing, making it possible to pass more than one texture in the vertex structure.

5.4.2.14 Cubic Environment Mapping

Environment maps allow applications to render scenes with complex lighting and reflections while significantly decreasing CPU load. There are several methods to generate environment maps such as spherical, circular, and cubic. The Intel 835 chipset GMCH has chosen to support cubic reflection mapping over spherical and circular because it is the best choice to provide real-time environment mapping for complex lighting and reflections.

Cubic mapping requires a texture map for each of the six cube faces. Pointing a camera with a 90-degree field-of-view in the appropriate direction can generate these. Per-vertex vectors (normal, reflection or refraction) are interpolated across the polygon and the intersection of these vectors with the cube texture faces is calculated. Texel values are read from the intersection point on the appropriate face and filtered accordingly.

5.4.2.15 Bump Mapping

Bump mapping is a feature in the Intel 835 chipset GMCH that enables a surface to appear wrinkled or dimpled without the need to model these depressions geometrically. By perturbing environment map texture coordinates on a per-pixel basis using delta values read from the bump map, non-uniform lighting effects (reflections, etc.) can be applied. This can give flat objects a bumpy or raised appearance. Embossing, a simpler form of bump mapping, is achieved by layering two identical texture maps. It can be supported through software to give the appearance of depth.

5.4.3 Raster Engine

The raster engine is where the color data such as fogging, specular RGB, texture map blending, and so forth are processed. The final color of the pixel is calculated and the RGBA value combined with the corresponding components resulting from the texture engine. These textured pixels are modified by the specular and fog parameters. These specular highlighted, fogged, textured pixels are color blended with the existing values in the frame buffer. In parallel, stencil, alpha and depth buffer tests are conducted which determines whether the frame and depth buffers are updated with the new pixel values.

5.4.3.1 Texture Map Blending

Multiple textures can be blended together in an iterative process and applied to a primitive. The Intel 835 chipset GMCH allows up to four distinct or shared texture coordinates and texture maps to be specified onto the same polygon. GMCH supports using a texture coordinate set to access multiple texture maps. State variables in multiple textures are bound to texture coordinates, texture map or texture blending.

5.4.3.2 Combining Intrinsic and Specular Color Components

The Intel 835 chipset GMCH allows an independently specified and interpolated "specular RGB" attribute to be added to the post-texture blended pixel color. This feature provides a full RGB specular highlight to be applied to a textured surface, permitting a high quality reflective colored lighting effect not available in devices, which apply texture after the lighting components have been combined. If the specular-add state variable is disabled, only the resultant colors from the map blending are used. If this state variable is enabled, RGB values from the output of the map blending are added to values for RS, GS, and BS on a component-by-component basis.

5.4.3.3 Color Shading Modes

The raster engine supports the flat and Gouraud shading modes. These shading modes are programmed by the appropriate state variables issued through the command stream.

Flat shading is performed by smoothly interpolating the vertex intrinsic color components (Red, Green, Blue), specular (RGB), fog, and alpha to the pixel, where each vertex color has the same value. The set-up engine substitutes one of the vertex's attribute values for the other two vertices' attribute values thereby creating the correct flat shading terms. This condition is set up by the appropriate state variables issued prior to rendering the primitive.

Gouraud shading is performed by smoothly interpolating the vertex intrinsic color components (Red, Green, Blue). specular (RGB), fog, and alpha to the pixel, where each vertex color has a different value.

All the attributes can be selected independently to one of the shading modes by setting the appropriate value state variables.

5.4.3.4 Color Dithering

Color dithering in the Intel 835 chipset GMCH helps to hide color quantization errors. Color dithering takes advantage of the human eye's propensity to "average" the colors in a small area. Input color, alpha, and fog components are converted from 8-bit components to 5-bit or 6-bit components by dithering. Dithering is performed on blended textured pixels. In 32-bit mode, dithering is not performed on the components.

5.4.3.5 Vertex and Per-Pixel Fogging

Fogging is used to create atmospheric effects such as low visibility conditions in flight simulator-type games. It adds another level of realism to computer-generated scenes. Fog can be used for depth cueing or hiding distant objects. With fog, distant objects can be rendered with less detail (fewer polygons), thereby improving the rendering speed or frame rate.

Fog is simulated by attenuating the color of an object with the fog color as a function of distance, and the greater the distance, the higher the density (lower visibility for distant objects). There are two ways to implement the fogging technique: per-vertex (linear) fogging and per-pixel (non-linear) fogging. The per-vertex method interpolates the fog value at the vertices of a polygon to determine the fog factor at each pixel within the polygon. This method provides realistic fogging as long as the polygons are small. With large polygons (such as a ground plane depicting an airport runway), the per-vertex technique results in unnatural fogging

The Intel 835 chipset GMCH supports both types of fog operations, vertex and per-pixel or table fog. If fog is disabled, the incoming color intensities are passed unchanged to the destination blend unit.

If fog is enabled, the incoming pixel color is blended with the fog color based on a fog coefficient on a per-pixel basis before sending to the destination blend unit.

5.4.3.6 Alpha Blending (Frame Buffer)

Alpha blending in the Intel 835 chipset GMCH adds the material property of transparency or opacity to an object. Alpha blending combines a source pixel color (RsGsBs) and alpha (As) component with a destination pixel color (RdGdBd) and alpha (Ad) component. For example, this is so that a glass surface on top (source) of a red surface (destination) would allow much of the red base color to show through.

Blending allows the source and destination color values to be multiplied by programmable factors and combined using a programmable blend function. The combined and independent selection of factors and blend functions for color and alpha is supported.

5.4.3.7 Color Buffer Formats: (Destination Alpha)

The raster engine supports 8-bit, 16-bit, and 32-bit color buffer formats. The 8-bit format is used to support planar YUV420 format, which used only in motion compensation and arithmetic stretch format. The bit format of color and Z are allowed to mix.

The Intel 835 chipset GMCH supports an 8-bit destination alpha in 32-bit mode. Destination alpha is supported in 16-bit mode in 1555 or 4444 format.

The Intel 835 chipset GMCH does not support general 3D rendering to 8-bit surfaces. Eight-bit destinations are supported for operations on planar YUV surfaces (e.g., stretch BLTs) where each 8-bit color component is written in a separate pass. GMCH also supports a mode where both U and V planar surfaces can be operated on simultaneously.

The frame buffer of GMCH contains at least two hardware buffers-the front buffer (display buffer) and the back buffer (rendering buffer). While the back buffer may actually coincide with (or be part of) the visible display surface, a separate (screen or window-sized) back buffer is typically used to permit double-buffered drawing. That is, the image being drawn is not visible until the scene is complete and the back buffer made visible or copied to the front buffer through a 2D BLT operation. Rendering to one buffer and displaying from the other buffer removes image-tearing artifacts. Additionally, more than two back buffers (e.g., triple-buffering) can be supported.

5.4.3.8 Depth Buffer

The raster engine is able to read and write from this buffer and use the data in per-fragment operations that determine whether resultant color and depth value of the pixel for the fragment are updated or not.

Typical applications for entertainment or visual simulations with exterior scenes require far/near ratios of 1000 to 10000. At 1000, 98 percent of the range is spent on the first 2 percent of the depth. This can cause hidden surface artifacts in distant objects, especially when using 16-bit depth buffers. A 24-bit Z-buffer provides 16 million Z-values as opposed to only 64K with a 16-bit Z-buffer. With lower Z-resolution, two distant overlapping objects may be assigned the same Z-value. As a result, the rendering hardware may have a problem resolving the order of the objects, and the object in the back may appear through the object in the front.

By contrast, when W (or eye-relative Z) is used, the buffer bits can be more evenly allocated between the near and far clip planes in world space. The key benefit is that the ratio of far and near is no longer an issue, allowing applications to support a maximum range of miles, yet still get reasonably accurate depth buffering within inches of the eye point. The selection of depth buffer size is relatively independent of the color buffer. A 16-bit Z/W or 24-bit Z/W buffer can be selected with a 16-bit color buffer. Z buffer is not supported in 8-bit mode.

5.4.3.9 Stencil Buffer

The raster engine provides 8-bit stencil buffer storage in 32-bit mode and the ability to perform stencil testing. Stencil testing controls 3D drawing on a per-pixel basis, which conditionally eliminates a pixel on the outcome of a comparison between a stencil reference value and the value in the stencil buffer at the location of the source pixel being processed. They are typically used in multipass algorithms to achieve special effects, such as decals, outlining, shadows and constructive solid geometry rendering.

One of three possible stencil operations is performed when stencil testing is enabled. The stencil operation specifies how the stencil buffer is modified when a fragment passes or fails the stencil test. The selection of the stencil operation is based upon the result of the stencil test and the depth test. A stencil write mask is included that controls the writing of particular bits into the stencil buffer. It chooses between the destination value and the updated value on a per-bit basis. The mask is 8 bits wide.

5.4.3.10 Projective Textures

The Intel 835 chipset GMCH supports two simultaneous projective textures at full rate processing. These textures require 3 floating-point texture coordinates to be included in the FVF format. Projective textures enable special effects such as projecting spotlight textures obliquely onto walls and other surfaces.

5.4.4 2D Engine

The Intel 835 chipset GMCH provides an extensive set of 2D instructions and 2D hardware acceleration for block transfers of data (BLTs). The BLT engine provides the ability to copy a source block of data to a destination and perform operations (e.g., ROP1, ROP2, and ROP3) on the data using a pattern, and/or another destination. The stretch BLT engine is used to move source data to a destination that need not be the same size, with source transparency. Performing these common tasks in hardware reduces CPU load, and improves performance.

5.4.4.1 256-Bit Pattern Fill and BLT Engine

Use of this BLT engine accelerates the Graphical User Interface (GUI) of Microsoft Windows*. The GMCH BLT engine provides hardware acceleration of block transfers of pixel data for many common Windows operations. The term BLT refers to a block transfer of pixel data between memory locations. The BLT engine can be used for the following:

- Moving rectangular blocks of data between memory locations
- Aligning data
- Performing logical operations (raster ops)

The rectangular block of data does not change as it is transferred between memory locations. Data to be transferred can consist of regions of memory, patterns, or solid color fills. A pattern is always 8x8 pixels wide and may be 8, 16, or 32 bits per pixel.

The Intel 835 chipset GMCH BLT engine has the ability to expand monochrome data to a color depth of 8, 16, or 32 bits. BLTs can be either opaque or transparent. Opaque transfers move the specified data to a destination. A transparent transfer compares the color of destination and source and writes according to the mode of transparency selected.

Data is horizontally and vertically aligned at the destination. If the destination for the BLT overlaps with the source memory location, GMCH can specify which area in memory to begin the BLT transfer. Hardware is included for all 256 raster operations (source, pattern, and destination) defined by Microsoft, including transparent BLT.

GMCH has instructions to invoke BLT operations, permitting software to set up instruction buffers and use batch processing. GMCH can perform hardware clipping during BLTs.

5.4.4.2 Alpha Stretch BLT

The stretch BLT function can stretch source data in the X and Y directions to a destination larger or smaller than the source. Stretch BLT functionality expands a region of memory into a larger or smaller region using replication and interpolation. The stretch BLT function also provides format conversion and data alignment.

5.4.5 Overlay Plane

The overlay engine provides a method of merging either video capture data (from an external video capture device) or data delivered by the CPU, with the graphics data on the screen.

5.4.5.1 Multiple Overlays

A single overlay plane and scalar is implemented. This overlay plane can be connected to the primary display. The overlay may be displayed in a multi-monitor scenario for single-pipe simultaneous displays only.

Picture-in-picture feature is supported using software through the arithmetic stretch blitter.

5.4.5.2 Source/Destination ColorKeying/ChromaKeying

Overlay source/destination chromakeying enables blending of the overlay with the underlying graphics background. Destination color-/chroma-keying can be used to handle occluded portions of the overlay window on a pixel-by-pixel basis that is actually an underlay. Destination colorkeying supports a specific color (8- or 15-bit) mode as well as 32-bit alpha blending.

Source color-/chroma-keying is used to handle transparency based on the overlay window on a pixel-by-pixel basis. This is used when “blue screening” an image to overlay the image on a new background later.

5.4.5.3 Gamma Correction

To compensate for overlay color intensity loss, the overlay engine supports independent gamma correction. This allows the overlay data to be converted to linear data or corrected for the display device when not blending.

5.4.5.4 Display Source Color Formats

Display units have a source color format and a destination color format. There may be an additional color format that is an intermediate color format. Source color formats are the formats that are resident in memory. The destination format is the one that makes its way out of the device intending to be displayed on a display device. Intermediate formats are used when combining pixels from different sources through blending.

5.4.5.5 Color Control

Color control provides a method of changing the color characteristics of the pixel data. It is applied to the data while in YUV format and uses input parameters such as brightness, saturation, hue (tint) and contrast. This feature is supplied only for the overlay and works only in YUV formats.

5.4.5.6 X/Y Mirroring

Both X or Y mirroring in the overlay is supported for video conferencing applications.

5.4.5.7 Dynamic Bob and Weave

Interlaced data that originates from a video camera creates two fields that are temporally offset by 1/60 of a second. There are several schemes to de-interlace the video stream: line replication, vertical filtering, field merging, and vertical temporal filtering. Field merging takes lines from the previous field and inserts them into the current field to construct the frame – this is known as *weaving*. This is the best solution for images with little motion; however, showing a frame that consists of the two fields when there is motion in the scene produces serration or feathering of moving edges. Vertical filtering or “bob” interpolates adjacent lines rather than replicating the nearest neighbor. This is the best solution for images with motion, however, it has reduced spatial resolution in areas that have no motion and introduces jaggies. In absence of any other de-interlacing, these form the baseline and are supported by the Intel 835 chipset GMCH.

5.4.6 Video Functionality

5.4.6.1 MPEG-2 Decoding

GMCH MPEG2 decoding supports hardware Motion Compensation (MC).

GMCH can accelerate video decoding for the following video coding standards:

- MPEG-2: Full feature support
- MPEG-1: Full feature support
- H.261: Full feature support
- H.263: Full feature support
- H.263+: Most of features with some exceptions of H.263+ optional features

- MPEG-4: Only supports some features in the simple profile.

5.4.6.2 Hardware Motion Compensation

The Motion Compensation (MC) process consists of reconstructing a new picture by predicting (either forward, backward, or bi-directional) the resulting pixel colors from one or more reference pictures. The Intel 835 chipset GMCH receives the video stream and implements motion compensation and subsequent steps in hardware. Performing motion compensation in hardware reduces the processor demand of software-based MPEG-2 decoding and improves system performance.

5.5 Internal Graphics Display Interface

The GMCH has two dedicated display ports, the analog port and digital display port A (RGBA). Each display port has control signals that may be used to control, configure and/or determine the capabilities of an external device. The data that is sent out the display port comes from pipe A.

GMCH's digital display ports are capable of driving a 81.23-MHz pixel clock. The RGBA interface can support a variety of TV encoders, external DACs, LVDS transmitters, and TMDS transmitters. The display port has control signals that may be used to control, configure and/or determine the capabilities of an external device. The data that is sent out the display port comes from pipe A.

5.5.1 Analog Display Port Characteristics

The analog display port provides a RGB signal output along with a HSYNC and VSYNC signal. There is an associated DDC signal pair that is implemented using GPIO pins dedicated to the analog port. The intended target device is for a CRT-based monitor with a VGA connector.

5.5.1.1 Integrated RAMDAC

The display function contains a RAM-based Digital-to-Analog Converter (RAMDAC) that transforms the digital data from the graphics and video subsystems to analog data for the CRT monitor. Three 8-bit DACs provide the R, G, and B signals to the monitor.

5.5.1.2 DDC (Display Data Channel)

DDC is defined by VESA. Its purpose is to allow communication between the host system and display. Both configuration and control information can be exchanged allowing plug-and-play systems to be realized. Support for DDC 1 and 2 is implemented.

5.5.2 RGBA Display Interface

The Intel 835 chipset GMCH has one option for driving digital displays. The Intel 835 chipset GMCH contains a dedicated digital display channel that can support RGBA devices.

5.5.2.1 Dedicated Digital Display Channel - RGBA

The Intel 835 chipset GMCH has a dedicated port for digital display support. It consists of a 12-bit digital data bus with accompanying clocks and control signals, and a 4-bit digital alpha data bus. Refer to Section 3.2.1 for a detailed description of these signals. This port utilizes a 1.5-V interface for high speed signaling, supporting a pixel clock up to 81.23 MHz. The port is designed to connect to a transmission device capable of TMDS or TV-out type signaling.

5.5.2.2 Interlaced Timing Details

CE devices use the relative placement of the VSYNC and HSYNC timing signals to discern odd and even field timing. For odd field detection, or the first field, the leading edge of VSYNC is within a ±31-Pclk coincident window around the leading edge of HSYNC. Preference is for the leading edge of VSYNC to be coincident with the leading edge HSYNC. If the leading edge of VSYNC is outside of the coincident window, then an even field, or the second field, is detected. The polarity of HSYNC and VSYNC are both programmable.

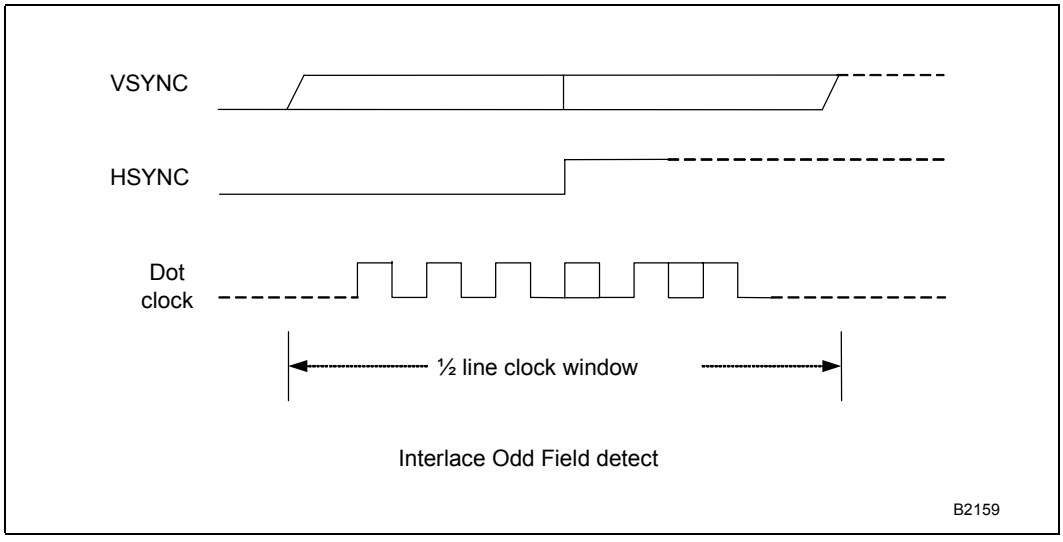
VSYNC transitions are within ½ line of HSYNC - Field 1 Otherwise - Field 2

For 525 line interlaced operation:
Transition in the window defines line 1 (first line of field 1).
Transition outside the window is line 263.

For 625 line interlaced operation:
Transition in the window defines line 1 (first line of field 1).
Transition outside the window is line 313.

For 1080 line interlaced operation:
Transition in the window defines line 1 (first line of field 1).
Transition outside the window is line 563.

Figure 12. Interlaced Timing Details Using HSYNC and VSYNC for Odd/Even Field Downstream Detection



5.5.2.3 Interlaced Video Field display

The current product requirements for interlaced timing support are the following:

- Interlace support for plane A graphics plane.
- Interlace support for video overlay window.

5.5.2.4 DDC (Display Data Channel)

The dedicated digital display interface (RGBA) uses the M_GPIO0/M_GPIO1 or M_GPIO3/M_GPIO2 to interrogate the panel. GMCH supports the DDC2B protocol to initiate the transfer of EDID information. The dedicated digital display interface uses the I2C bus to interrogate the external transmitter.

5.5.2.5 TV Encoder Capabilities

The Intel 835 chipset GMCH supports TV encoders through the RGBA interface. GMCH generates the proper timing for the external encoder. The external encoder is responsible for generation of the proper format signal. Because the GMCH RGBA interface is 1.5 V, care should be taken to ensure that the TV encoder is operational at that signaling voltage.

The TV-out interface on GMCH is addressable as a master device. This allows an external TV encoder device to drive a pixel clock signal on RGBA_CLKIN that the Intel 835 chipset uses as a reference frequency. The frequency of this clock is dependent on the output resolution required. Data is driven to the encoder across 12 data lines, along with clock pair and sync signals. The encoder can expect a continuous flow of data from GMCH because data is not throttled.

5.5.2.6 Flicker Filter and Overscan Compensation

Overscan compensation scaling and the flicker filter are done in the external TV encoder chip. Care must be taken to allow for support of TV sets with high performance de-interlacers and progressive scan displays connected by way of a non-interlaced signal. Timing is generated with pixel granularity to allow more overscan ratios to be supported.

5.5.2.7 Analog Content Protection

Analog content protection is provided through the external encoder using Macrovision* 7.01. DVD software must verify the presence of a Macrovision TV encoder before playback continues. Simple attempts to disable the Macrovision operation must be detected.

5.5.2.8 Support of Progressive Scan SDTV TVs

Support is included for progressive scan TV devices through the TV port. This support includes the resolutions of 480p and 1080i using both a YUV analog signal with sync on Y and a RGB HV connection for TVs with that connection.

5.5.3 RGBA Port Configuration

The data on the RGBA port is formatted as described in Table 88:

Table 88. RGBA Data Mapping

RGBA Data	PL	PH
RGBA_ALPHA3	α[3]	α[7]
RGBA_ALPHA2	α[2]	α[6]
RGBA_ALPHA1	α[1]	α[5]
RGBA_ALPHA2	α[0]	α[4]
RGBA_DATA11	G[3]	R[7]
RGBA_DATA10	G[2]	R[6]
RGBA_DATA9	G[1]	R[5]
RGBA_DATA8	G[0]	R[4]
RGBA_DATA7	B[7]	R[3]
RGBA_DATA6	B[6]	R[2]
RGBA_DATA5	B[5]	R[1]
RGBA_DATA4	B[4]	R[0]
RGBA_DATA3	B[3]	G[7]
RGBA_DATA2	B[2]	G[6]
RGBA_DATA1	B[1]	G[5]
RGBA_DATA0	B[0]	G[4]

The data is double-pumped with PL on the first clock edge and PH in the second clock edge.

5.6 ARIB Support

5.6.1 ARIB TR-B15 Guideline

The Intel 835 chipset specifically supports TR-B15 operational guidelines for digital satellite broadcasting (detailed implementation guideline for receiver) of the ARIB specification.

5.6.2 ARIB Resolution Support

The Intel 835 chipset supports the resolutions outlined in Table 89.

Table 89. ARIB TR-B15 Plane Resolutions (Sheet 1 of 2)

Plane Name	Requirements	
Still Picture	Resolution	1920x1080x16Y, CbCr (4:2:2), 16:9
		720x480x16, YCbCr (4:2:2), 16:9
		720x480x16, YCbCr (4:2:2), 4:3
Text and Graphic	Resolution	960x540x8, 16:9 (Display resolution is 1920x1040 – 1 pixel on the plane is transferred to 2x2 pixel on display)
		720x480x8, 16:9
		720x480x8, 4:3

Table 89. ARIB TR-B15 Plane Resolutions (Sheet 2 of 2)

Plane Name	Requirements	
Superimpose Text	Resolution	960x540x8m 16:9 (Display resolution is 1920x1080 – 1 pixel on the plane is transferred to 2x2 pixel on display)
		720x480x8, 16:9
		720x480x8, 4:3

5.6.3 Overlay Display

The overlay plane is displayed by means of color keying only. In order to display the overlay plane, a color must be used to key the overlay image into the display.

5.7 GMCH Power and Thermal Management

The following list provides the GMCH power and thermal management features:

- ACPI 1.0b and 2.0 support
- System states: S0, S1, S3, S4, S5
- CPU states: C0, C1, C2 (Desktop)
- Graphics states: D0, D1, D3
- Compatible with Intel 815EM AGP busy/stop protocol
- Thermal throttling for main memory, and graphics

5.7.1 ACPI 2.0 Support

Advanced Configuration and Power Management Interface (ACPI) primarily describes and runs motherboard devices. It is completely controlled by the operating system that OS drivers directly power down PCI devices. System or SMI BIOS plays a part of waking the system, however. Device drivers save and restore state while bus drivers change the physical power state of the device.

The Intel 835 chipset GMCH power management architecture is designed to allow single systems to support multiple suspend modes and to switch between those modes as required. A suspended system can be resumed via a number of different events. The system returns to full operation where it can continue processing or be placed into another suspend mode (potentially a lower power mode than it resumed from).

GMCH supports the minimum requirements for ACPI support. GMCH must support the minimum requirements for both system logic and for graphics controllers, as well as be capable of controlling monitors' minimum functions. The transition sequences of entering and exiting system, CPU and graphics states are described in respective sections below.

5.7.2 ACPI States Supported

5.7.2.1 ACPI Supported States

When an internal graphics device is used, the Intel 835 chipset supports the following ACPI states:

- System states
  - **G0/S0** Full on
  - **G1/S1** Power On Suspend (POS). System context preserved
  - **G1/S3** Suspend to RAM (STR). Power and context lost to chipset
  - **G1/S4** Suspend to Disk (STD). All power lost (except wake-up on ICH4)
  - **G2/S5** Hard off. Total reboot
- CPU states
  - **C0** Full on
  - **C1** Auto halt
  - **C2** Stop Grant
  - Clock to CPU stopped clock to CPU stopped or CPU DPSLP# pin asserted
- Internal graphics (IGD) states:
  - **D0** Full on, display active
  - **D1** Low power state, low latency recovery
  - **D3Hot** All state lost other than PCI configuration. Memory lost (optionally)
  - **D3Cold** Power off

5.7.3 Intel® 835 Chipset System and CPU States

Table 90 shows the state combinations that the Intel 835 chipset supports.

Table 90. Intel® 835 Chipset System and CPU States

Global (G) State	Sleep (S) State	CPU (C) State	Processor State	Description
G0	S0	C0	Full On	Full On
G0	S0	C1	Auto-Halt	Auto Halt
G0	S0	C2	Stop Grant	Stop Grant
G1	S3	Power off	Power off	Suspend to RAM
G1	S4	Power off	Power off	Suspend to Disk
G2	S5	Power off	Power off	Hard Off
G3	NA	Power off	Power off	Mechanical Off

5.7.4 Intel® 835 Chipset Family CPU "C" States

5.7.4.1 Full-On (C0)

This is the only state that runs software. All clocks are running, STPCLK# is deasserted and the processor core is active. The processor can service snoops and maintain cache coherency in this state.

5.7.4.2 Auto-Halt (C1)

The first level of power reduction occurs when the processor executes an auto-halt instruction. This stops the execution of the instruction stream and greatly reduces the processors power consumption. The processor can service snoops and maintain cache coherency in this state.

5.7.4.3 Stop Grant (C2)

The next level of power reduction occurs when the processor is placed into the Stop Grant state by the assertion of STPCLK#. The GMCH supports only the Stop Grant state in C2.

5.7.5 System "S" States

5.7.5.1 Powered-On-Suspend (POS) (S1)

The deepest level of power savings that can be achieved by only shutting down clocks occurs in the S1 state. The only clock remaining active in the system in the S1 state is the RTC clock. This clock is used to detect wake events and to run the hardware in the resume well in the ICH4 used to reactivate the system.

During the S1 state the CPU and the Intel 835 chipset GMCH power is on, however there is no activity, so the only power consumed is the leakage power. The clock synthesizer) is powered off, this shuts the clocks off in the host, memory, and I/O clock groups. If the D1 state is used for internal graphics, a clock must be provided to GMCH for DPMS signaling to the CRT.

5.7.5.2 Suspend-To-RAM (STR) (S3)

The final level of power savings for the Intel 835 chipset GMCH is achievable when the host clock, memory group, and I/O clock group clocks are shutdown and the GMCH is powered down. This occurs when the system transitions to the S3 state. During transition to the S3 state, first the STPCLK# is asserted and the stop grant cycle snooped by the GMCH and forwarded over hub interface where it is received by the ICH4. At this point the GMCH is functioning in the C2 state. The GMCH places all of the SDRAM components into the self-refresh mode. After the GMCH has placed all of the SDRAM components in self-refresh, it is safe to enter the STR state. The ICH4 then asserts a signal, SLP_S1#, to the clock synthesizer to shutdown all of the clocks in the host and memory clock groups.

The GMCH assumes that no hub interface cycle (except special cycles) occurs while the GMCH is in the C3 state. The processor cannot snoop its caches to maintain coherency while in the C3 state.

GMCH contains no isolation circuitry and MUST be powered down once STR is reached. If GMCH is powered up and driving outputs to devices that are powered down, component damage results.

5.7.5.3 S4 (SUSPEND TO DISK), S5 (Soft Off) State

The Intel 835 chipset does not distinguish between suspend to ram (S3), suspend to disk (S4) and soft off (S5) states. From the 835 chipset perspective, entry and exit to S4 or S5 states is the same as entry and exit to S3 state.

5.7.6 Internal Graphics "D" States

PC9x implies that D0 and D3 are obligatory for graphics controllers. D0, D2, and D3 are obligatory for monitors. The Intel 835 chipset GMCH also implements D1 for the graphics controller and monitors. System SDRAM state is generally controlled by S-states and C-states rather than D-states. With internal graphics the system SDRAM remains available when the CPU is in C3.

5.7.6.1 D0 Graphics Adapter State – Active State

In the D0 power state, everything is operating. This is the normal ON state for the ICD graphics functions. The GMCH graphics functions enter this state out of power-on-reset.

5.7.6.2 The D1 Graphics Adapter State

In the D1 power state, the graphics must go to a lower power state. The displays are blank, but memory and registers must be maintained. The emphasis is on a fast recovery in this mode.

5.7.6.3 The D3 Graphics Adapter State

The D3 power state is the lowest power mode. Displays are off, and the registers and memory need not be maintained. The PCI config space must be accessible, in order to write the power state back to D0.

When the OS decides to put the IGD graphics functions into D3 power state, it calls the IGD graphics driver so that the driver saves the device context. Device context consists of the IGD graphics mode as well as non-local video memory context. External parts context must also be stored.

5.7.6.4 Monitor [Analog CRT] States

The monitor is considered a child device of the graphics controller. Its ACPI states are controlled through the graphics controller. Display Power Management Signaling (DPMS) is a Video Electronics Standards Association (VESA) specification that provides a method for the graphics controller to put the monitor in a particular power management state by controlling the presence or absence of pulses on the HSYNC and VSYNC signals. The D state of the CRT monitor can be set independently of the graphics controller, but is always equal to or higher (in number, lower in power) than the graphics controller. The monitor is considered a "child device" to the graphics controller. Table 91 lists each combination.

Table 91. Combinations of CRT and Graphics Power Down States (Sheet 1 of 2)

Graphics Controller	CRT State	HSYNC/VSYNC Status
D0	D0 = On	Pulse HSYNC and VSYNC
D0	D1 = Standby	Pulse VSYNC
D0	D2 = Suspend	Pulse HSYNC

Table 91. Combinations of CRT and Graphics Power Down States (Sheet 2 of 2)

D0	D3 = Off	No pulse on HSYNC and VSYNC
D1	D1	Pulse VSYNC
D1	D2	Pulse HSYNC
D1	D3	No pulse on HSYNC and VSYNC
D3	D3	No pulse on HSYNC and VSYNC

In D1 state, the graphics controller must be able to toggle either HSYNC or VSYNC, depending on the CRT state.

5.7.6.5 DPMS Clock Signaling in S1 (D1) State

When the Intel 835 chipset GMCH graphics controller is in the D1 state, the graphics core clock and dot clocks are stopped, causing HSYNC and VSYNC generation to stop. If the system is configured to allow the graphics controller to be in D1 while the system is in mobile S1, all clocks in the system, including the clock generator chip are shut off. Potentially the only clock running is the 32 kHz of the real time clock. The DPMS clock signal is muxed with GAD30 to provide clock source to generate pulses on HSYNC and VSYNC in the D1 state. The DPMS clock signal requires an external clock source, which may be 32 kHz or a 33/66 MHz clock. DPMS_CLK is NOT required if S1 states is NOT supported.

5.7.7 System Memory Dynamic SMCKE Support

To reduce EMI and preserve battery life, clocks to unpopulated DIMMs are turned off. The DRB registers are read to determine if the row is populated. Clocks are turned off in pairs because SMCLK[1:0] go to one DIMM, SMCLK[3:2] go to another DIMM. The main memory SDRAMs are power managed during normal operation and in low power modes. Each row has a separate SMCKE (clock enable) pin that is used for power management. SMCKE is used to put the SDRAM rows into power down mode. Active power management is employed during normal operation. The memory setting is determined by the thermals of the system and the number of chips in a row. Following refresh, all SDRAMs are powered down except the one for which there is the first pending request, if any.

5.8 Thermal Management

With the addition of the integrated graphics device, passive heat dissipation may not be enough and active cooling may reach its limit. Counter-based throttling does not correlate well with the environment, especially in a small form factor where the outside ambient temperature varies greatly, along with internal conditions such as heavy 3D content.

The Intel 835 chipset GMCH has several methods for monitoring and/or handling thermal issues. GMCH contains an on-die thermal sensor used for emergency throttling and shutdown. GMCH contains a bandwidth monitor on the IGD and the SDRAM interfaces. If the bandwidth exceeds a programmed amount, the GMCH automatically stalls to avoid thermal problems.

5.8.1 Thermal Sensor

The Intel 835 chipset GMCH has an on-die thermal sensor for emergency throttling and shutdown. A thermal sensor provides a closed-loop feedback path, and an emergency indicator.

5.8.2 Graphic Thermal Throttling

The Intel 835 chipset 3D engine contains a throttling mechanism between the 3D engine and the memory interface. In non-throttled-state, the 3D pipe has two signals that control the flow of data to and from the local cache. There are three programmable values for the 3D pipe duty cycle, of which one (at most) is in use at any given time.

5.8.3 System and Graphics Memory Bandwidth Monitoring and Throttling

The Intel 835 chipset has the capability for bandwidth monitoring/throttle mechanism for the system memory interface (applicable to the entire chipset family). If the counter window exceeds the bandwidth threshold, then the SDRAM throttling mechanism is invoked to limit the memory reads/writes to a lower bandwidth.

The bandwidth monitoring mechanism consists of a counter to measure SDRAM bandwidth being used. Depending on what is being monitored, reads and writes or both, a counter is incremented. If the number of read/writes during the monitoring period exceeds the value programmed, the throttling mechanism is invoked.

If the Intel 835 chipset GMCH detects an idle cycle where no traffic is encountered during the throttling window, the counter decrements and no throttling takes place. After the bandwidth reaches the determined bandwidth, the GMCH starts to throttle and continue throttling determined by the activity percentage. If the bandwidth never exceeds the set value, no throttling takes place. The GMCH exits the throttling mechanism and returns to monitoring traffic where the process starts over again.

5.9 Clocking

The Intel 835 chipset GMCH has the following clocks:

- 133-MHz, spread spectrum, low voltage differential HTCLK (HTCLKB) for processor side bus
- 66.666-MHz 3.3-V, spread spectrum, GB CLKOUT output clock for external hub/PCI buffer
- 66.666-MHz 3.3-V, spread spectrum, GB CLKIN0 from external buffer for hub interface
- 48-MHz, spread spectrum, 3.3-V DREF CLK for the display frequency syntheses (applicable only when internal graphics device is used)
- 13.5-81.23 MHz RGBA_CLKIN for TV encoder mode

The Intel 835 chipset has inputs for a low voltage, differential pair of clocks called HTCLK and HTCLKB. These pins receive a buffered host clock from the external clock synthesizer. This clock is used by all of the GMCH logic. This clock is also the reference clock for the graphics core PLL.

The graphic core and display interfaces are asynchronous to the rest of the GMCH. The graphics core runs at 100-165 MHz. The display PLLs uses the non-spread spectrum 48-MHz input to generate frequency range of 12-165 MHz.

5.10 XOR Test Chains

Another feature of the Intel 835 chipset is the support for XOR chain test modes. Product engineers use the XOR chain test mode during manufacturing and OEMs during board level connectivity tests. The main purpose of this test mode is to detect connectivity shorts between adjacent pins and to check proper bonding between I/O pads and I/O pins. There are 11 XOR test chains built into the chipset.

6.0 Intel® 835 Chipset Performance

The system performance for the Intel® 835 chipset GMCH described below is a breakdown of the data streams that complement the Mobile Intel® Celeron® processor. This section describes the overall performance of the GMCH. The following categories of performance are examined:

- CPU/835 chipset GMCH-M: Supports Mobile Intel Celeron processor
- System memory: Intel 835 chipset GMCH supports PC133 main memory
- RGBA only available with the Intel 835 chipset

Table 92. System Bandwidths

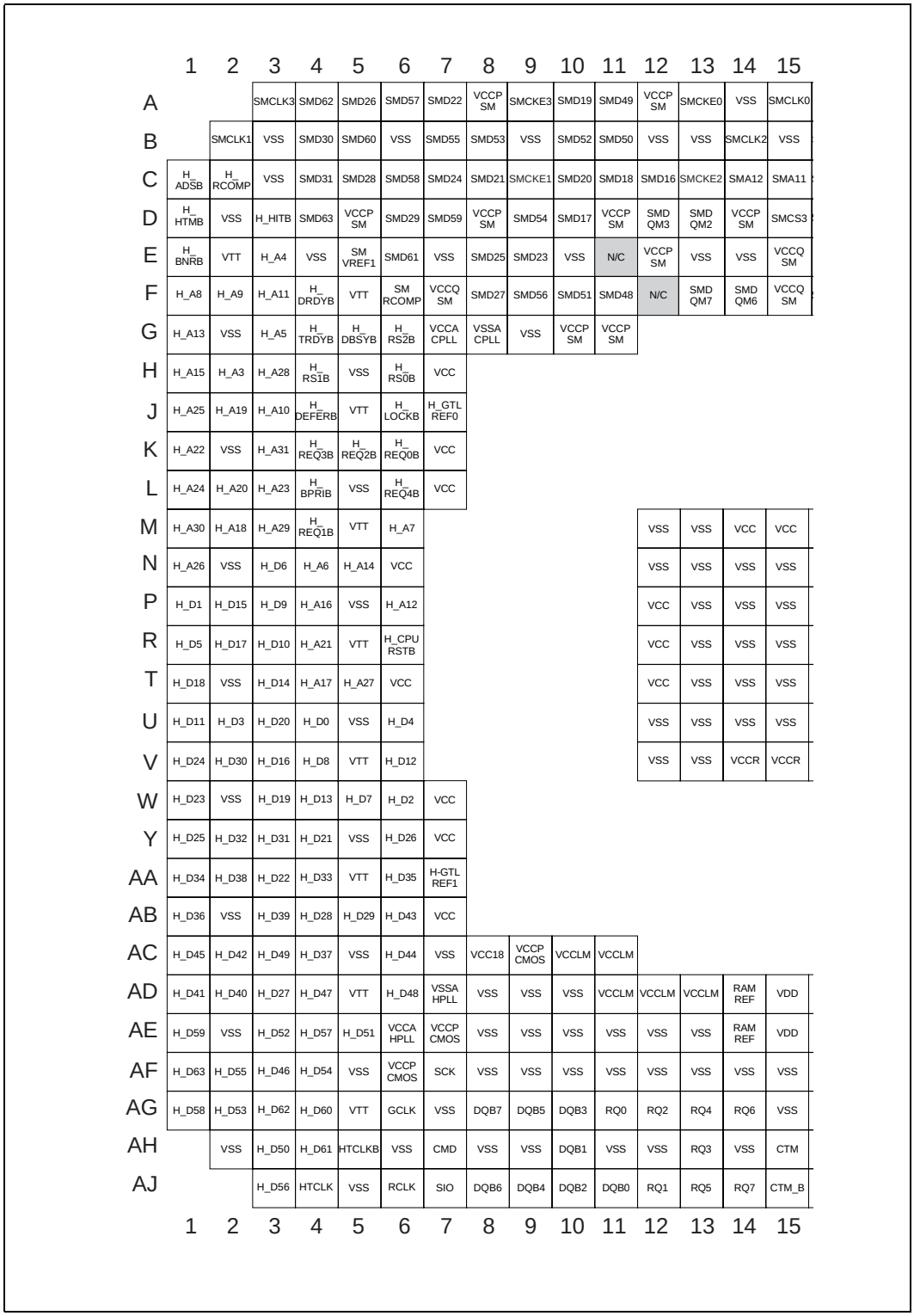
Interface	Clock Speed (MHz)	Samples Per Clock	Data Rate (Mega-samples/s)	Data Width (Bytes)	Bandwidth (Mbyte/s)
CPU Bus	133	1	133	8	1066
SDRAM	133	1	133	8	1064
RGBA	81.23	2	162.46	1.5	243.69
PCI 2.2	33	1	33	4	133

NOTE: Theoretical bandwidths only

7.0 Chipset Package Information

The following chapter provides the Intel® 835 chipset package and ballout information:

Figure 13. Intel® 835 Chipset Ballout Diagram (Left Side)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A			SMCLK3	SMD62	SMD26	SMD57	SMD22	VCCP SM	SMCKE3	SMD19	SMD49	VCCP SM	SMCKE0	VSS	SMCLK0
B		SMCLK1	VSS	SMD30	SMD60	VSS	SMD55	SMD53	VSS	SMD52	SMD50	VSS	VSS	SMCLK2	VSS
C	H_ ADSB	H_ RCOMP	VSS	SMD31	SMD28	SMD58	SMD24	SMD21	SMCKE1	SMD20	SMD18	SMD16	SMCKE2	SMA12	SMA11
D	H_ HTMB	VSS	H_HITB	SMD63	VCCP SM	SMD29	SMD59	VCCP SM	SMD54	SMD17	VCCP SM	SMD QM3	SMD QM2	VCCP SM	SMCS3
E	H_ BNRB	VTT	H_A4	VSS	SM VREF1	SMD61	VSS	SMD25	SMD23	VSS	N/C	VCCP SM	VSS	VSS	VCCQ SM
F	H_A8	H_A9	H_A11	H_ DRDYB	VTT	SM RCOMP	VCCQ SM	SMD27	SMD56	SMD51	SMD48	N/C	SMD QM7	SMD QM6	VCCQ SM
G	H_A13	VSS	H_A5	H_ TRDYB	H_ DBSYB	H_ RS2B	VCCA CPLL	VSSA CPLL	VSS	VCCP SM	VCCP SM				
H	H_A15	H_A3	H_A28	H_ RS1B	VSS	H_ RS0B	VCC								
J	H_A25	H_A19	H_A10	H_ DEFERB	VTT	H_ LOCKB	H_GTL REF0								
K	H_A22	VSS	H_A31	H_ REQ3B	H_ REQ2B	H_ REQ0B	VCC								
L	H_A24	H_A20	H_A23	H_ BPRIB	VSS	H_ REQ4B	VCC								
M	H_A30	H_A18	H_A29	H_ REQ1B	VTT	H_A7						VSS	VSS	VCC	VCC
N	H_A26	VSS	H_D6	H_A6	H_A14	VCC						VSS	VSS	VSS	VSS
P	H_D1	H_D15	H_D9	H_A16	VSS	H_A12						VCC	VSS	VSS	VSS
R	H_D5	H_D17	H_D10	H_A21	VTT	H_CPU RSTB						VCC	VSS	VSS	VSS
T	H_D18	VSS	H_D14	H_A17	H_A27	VCC						VCC	VSS	VSS	VSS
U	H_D11	H_D3	H_D20	H_D0	VSS	H_D4						VSS	VSS	VSS	VSS
V	H_D24	H_D30	H_D16	H_D8	VTT	H_D12						VSS	VSS	VCCR	VCCR
W	H_D23	VSS	H_D19	H_D13	H_D7	H_D2	VCC								
Y	H_D25	H_D32	H_D31	H_D21	VSS	H_D26	VCC								
AA	H_D34	H_D38	H_D22	H_D33	VTT	H_D35	H-GTL REF1								
AB	H_D36	VSS	H_D39	H_D28	H_D29	H_D43	VCC								
AC	H_D45	H_D42	H_D49	H_D37	VSS	H_D44	VSS	VCC18	VCCP CMOS	VCCLM	VCCLM				
AD	H_D41	H_D40	H_D27	H_D47	VTT	H_D48	VSSA HPLL	VSS	VSS	VSS	VCCLM	VCCLM	VCCLM	RAM REF	VDD
AE	H_D59	VSS	H_D52	H_D57	H_D51	VCCA HPLL	VCCP CMOS	VSS	VSS	VSS	VSS	VSS	VSS	RAM REF	VDD
AF	H_D63	H_D55	H_D46	H_D54	VSS	VCCP CMOS	SCK	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
AG	H_D58	H_D53	H_D62	H_D60	VTT	GCLK	VSS	DQB7	DQB5	DQB3	RQ0	RQ2	RQ4	RQ6	VSS
AH		VSS	H_D50	H_D61	HTCLKB	VSS	CMD	VSS	VSS	DQB1	VSS	VSS	RQ3	VSS	CTM
AJ			H_D56	HTCLK	VSS	RCLK	SIO	DQB6	DQB4	DQB2	DQB0	RQ1	RQ5	RQ7	CTM_B

Figure 14. Intel® 835 Chipset Ballout Diagram (Right Side)

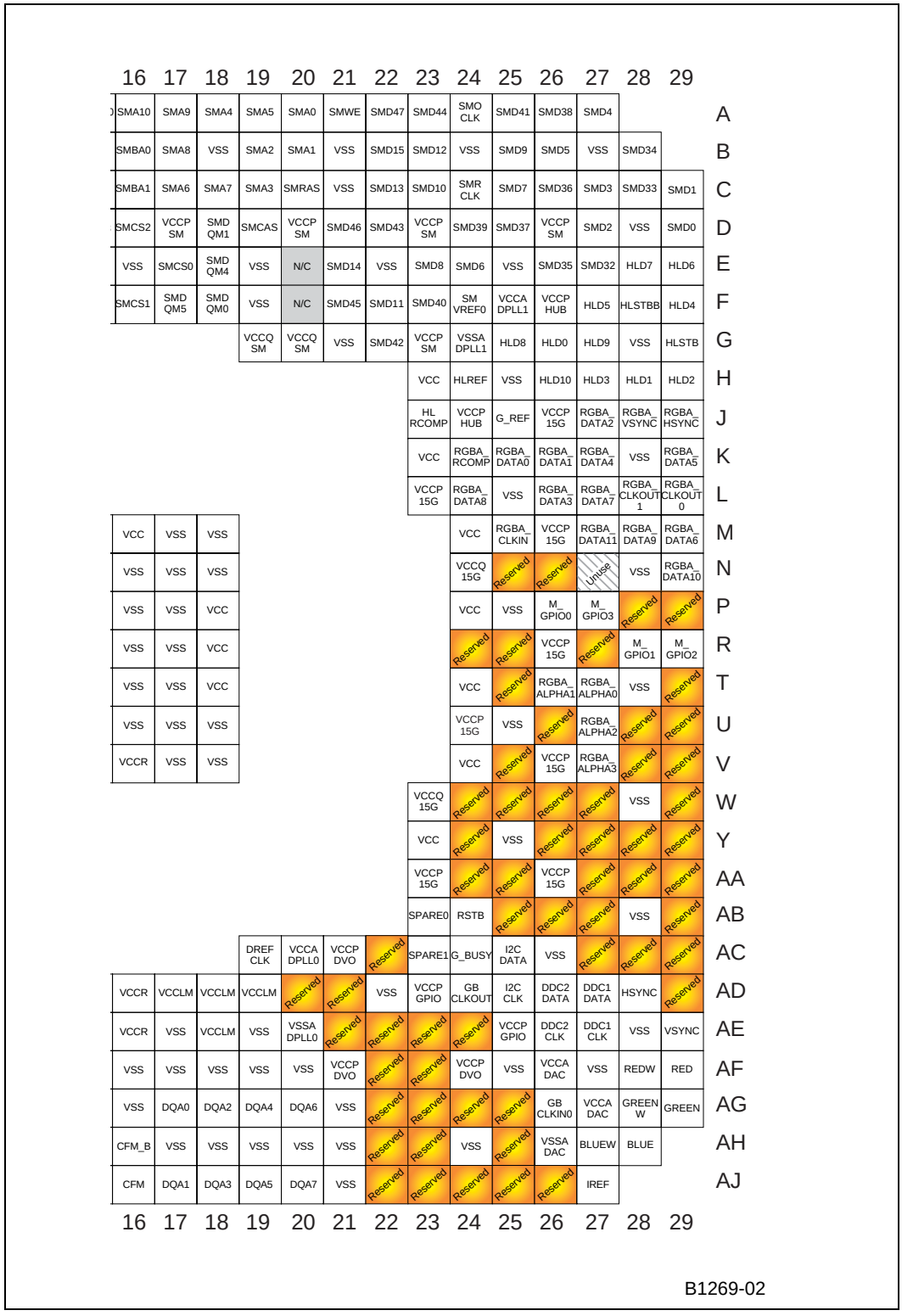


Table 93. Signal Name List (Signal Order) (Sheet 1 of 17)

Ball #	Signal Name
AH28	BLUE
AH27	BLUEW
AJ16	CFM
AH16	CFM_B
AH7	CMD
AH15	CTM
AJ15	CTM_B
AE27	DDC1 CLK
AD27	DDC1 DATA
AE26	DDC2 CLK
AD26	DDC2 DATA
AG17	DQA0
AJ17	DQA1
AG18	DQA2
AJ18	DQA3
AG19	DQA4
AJ19	DQA5
AG20	DQA6
AJ20	DQA7
AJ11	DQB0
AH1-	DQB1
AJ10	DQB2
AG10	DQB3
AJ9	DQB4
AG9	DQB5
AJ8	DQB6
AG8	DQB7
AC19	DREF CLK
AC24	G_BUSY
J25	G_REF
AG26	GB CLKIN0
AD24	GB CLKOUT
AG6	GCLK
AG29	GREEN
AG28	GREENW
H2	H_A3
E3	H_A4

Table 93. Signal Name List (Signal Order) (Sheet 2 of 17)

Ball #	Signal Name
G3	H_A5
N4	H_A6
M6	H_A7
F1	H_A8
F2	H_A9
J3	H_A10
F3	H_A11
P6	H_A12
G1	H_A13
N5	H_A14
H1	H_A15
P4	H_A16
T4	H_A17
M2	H_A18
J2	H_A19
L2	H_A20
R4	H_A21
K1	H_A22
L3	H_A23
L1	H_A24
J1	H_A25
N1	H_A26
T5	H_A27
H3	H_A28
M3	H_A29
M1	H_A30
K3	H_A31
C1	H_ADSB
E1	H_BNRB
L4	H_BPRIB
R6	H_CPU RSTB
U4	H_D0
P1	H_D1
W6	H_D2
U2	H_D3
U6	H_D4
R1	H_D5

Table 93. Signal Name List (Signal Order) (Sheet 3 of 17)

Ball #	Signal Name
N3	H_D6
W5	H_D7
V4	H_D8
P3	H_D9
R3	H_D10
U1	H_D11
V6	H_D12
W4	H_D13
T3	H_D14
P2	H_D15
V3	H_D16
R2	H_D17
T1	H_D18
W3	H_D19
U3	H_D20
Y4	H_D21
AA3	H_D22
W1	H_D23
V1	H_D24
Y1	H_D25
Y6	H_D26
AD3	H_D27
AB4	H_D28
AB5	H_D29
V2	H_D30
Y3	H_D31
Y2	H_D32
AA4	H_D33
AA1	H_D34
AA6	H_D35
AB1	H_D36
AC4	H_D37
AA2	H_D38
AB3	H_D39
AD2	H_D40
AD1	H_D41
AC2	H_D42

Table 93. Signal Name List (Signal Order) (Sheet 4 of 17)

Ball #	Signal Name
AB6	H_D43
AC6	H_D44
AC1	H_D45
AF3	H_D46
AD4	H_D47
AD6	H_D48
AC3	H_D49
AH3	H_D50
AE5	H_D51
AE3	H_D52
AG2	H_D53
AF4	H_D54
AF2	H_D55
AJ3	H_D56
AE4	H_D57
AG1	H_D58
AE1	H_D59
AG4	H_D60
AH4	H_D61
AG3	H_D62
AF1	H_D63
G5	H_DBSYB
J4	H_DEFERB
F4	H_DRDYB
J7	H_GTL REF0
AA7	H_GTL REF1
D3	H_HITB
D1	H_HITMB
J6	H_LOCKB
C2	H_RCOMP
K6	H_REQ0B
M4	H_REQ1B
K5	H_REQ2B
K4	H_REQ3B
L6	H_REQ4B
H6	H_RS0B
H4	H_RS1B

Table 93. Signal Name List (Signal Order) (Sheet 5 of 17)

Ball #	Signal Name
G6	H_RS2B
G4	H_TRDYB
J23	HL RCOMP
G26	HLD0
H28	HLD1
H29	HLD2
H27	HLD3
F29	HLD4
F27	HLD5
E29	HLD6
E28	HLD7
G25	HLD8
G27	HLD9
H26	HLD10
H24	HLREF
G29	HLSTB
F28	HLSTBB
AD28	HSYNC
AJ4	HTCLK
AH5	HTCLKB
AD25	I2C CLK
AC25	I2C DATA
AJ27	IREF
P26	M_GPIO0
R28	M_GPIO1
R29	M_GPIO2
P27	M_GPIO3
E11	N/C
E20	N/C
F12	N/C
F20	N/C
AD14	RAM REF
AE14	RAM REF
AJ6	RCLK
AF29	RED
AF28	REDW
AA24	RESERVED

Table 93. Signal Name List (Signal Order) (Sheet 6 of 17)

Ball #	Signal Name
AA25	RESERVED
AA27	RESERVED
AA28	RESERVED
AA29	RESERVED
AB25	RESERVED
AB26	RESERVED
AB27	RESERVED
AB29	RESERVED
AC22	RESERVED
AC27	RESERVED
AC28	RESERVED
AC29	RESERVED
AD20	RESERVED
AD21	RESERVED
AD29	RESERVED
AE21	RESERVED
AE22	RESERVED
AE23	RESERVED
AE24	RESERVED
AF22	RESERVED
AF23	RESERVED
AG22	RESERVED
AG23	RESERVED
AG24	RESERVED
AG25	RESERVED
AH22	RESERVED
AH23	RESERVED
AH25	RESERVED
AJ22	RESERVED
AJ23	RESERVED
AJ24	RESERVED
AJ25	RESERVED
AJ26	RESERVED
N25	RESERVED
N26	RESERVED
P28	RESERVED
P29	RESERVED

Table 93. Signal Name List (Signal Order) (Sheet 7 of 17)

Ball #	Signal Name
R24	RESERVED
R25	RESERVED
R27	RESERVED
T25	RESERVED
T29	RESERVED
U26	RESERVED
U28	RESERVED
U29	RESERVED
V25	RESERVED
V28	RESERVED
V29	RESERVED
W24	RESERVED
W25	RESERVED
W26	RESERVED
W27	RESERVED
W29	RESERVED
Y24	RESERVED
Y26	RESERVED
Y27	RESERVED
Y28	RESERVED
Y29	RESERVED
T27	RGBA_ALPHA0
T26	RGBA_ALPHA1
U27	RGBA_ALPHA2
V27	RGBA_ALPHA3
M25	RGBA_CLKIN
L29	RGBA_CLKOUT0
L28	RGBA_CLKOUT1
K25	RGBA_DATA0
K26	RGBA_DATA1
J27	RGBA_DATA2
L26	RGBA_DATA3
K27	RGBA_DATA4
K29	RGBA_DATA5
M29	RGBA_DATA6
L27	RGBA_DATA7
L24	RGBA_DATA8

Table 93. Signal Name List (Signal Order) (Sheet 8 of 17)

Ball #	Signal Name
M28	RGBA_DATA9
N29	RGBA_DATA10
M27	RGBA_DATA11
J29	RGBA_HSYNC
K24	RGBA_RCOMP
J28	RGBA_VSYNC
AG11	RQ0
AJ12	RQ1
AG12	RQ2
AH13	RQ3
AG13	RQ4
AJ13	RQ5
AG14	RQ6
AJ14	RQ7
AB24	RSTB
AF7	SCK
AJ7	SIO
F6	SM RCOMP
F24	SM VREF0
E5	SM VREF1
A20	SMA0
B20	SMA1
B19	SMA2
C19	SMA3
A18	SMA4
A19	SMA5
C17	SMA6
C18	SMA7
B17	SMA8
A17	SMA9
A16	SMA10
C15	SMA11
C14	SMA12
B16	SMBA0
C16	SMBA1
D19	SMCAS
A13	SMCKE0

Table 93. Signal Name List (Signal Order) (Sheet 9 of 17)

Ball #	Signal Name
C9	SMCKE1
C13	SMCKE2
A9	SMCKE3
A15	SMCLK0
B2	SMCLK1
B14	SMCLK2
A3	SMCLK3
E17	SMCS0
F16	SMCS1
D16	SMCS2
D15	SMCS3
D29	SMD0
C29	SMD1
D27	SMD2
C27	SMD3
A27	SMD4
B26	SMD5
E24	SMD6
C25	SMD7
E23	SMD8
B25	SMD9
C23	SMD10
F22	SMD11
B23	SMD12
C22	SMD13
E21	SMD14
B22	SMD15
C12	SMD16
D10	SMD17
C11	SMD18
A10	SMD19
C10	SMD20
C8	SMD21
A7	SMD22
E9	SMD23
C7	SMD24
E8	SMD25

Table 93. Signal Name List (Signal Order) (Sheet 10 of 17)

Ball #	Signal Name
A5	SMD26
F8	SMD27
C5	SMD28
D6	SMD29
B4	SMD30
C4	SMD31
E27	SMD32
C28	SMD33
B28	SMD34
E26	SMD35
C26	SMD36
D25	SMD37
A26	SMD38
D24	SMD39
F23	SMD40
A25	SMD41
G22	SMD42
D22	SMD43
A23	SMD44
F21	SMD45
D21	SMD46
A22	SMD47
F11	SMD48
A11	SMD49
B11	SMD50
F10	SMD51
B10	SMD52
B8	SMD53
D9	SMD54
B7	SMD55
F9	SMD56
A6	SMD57
C6	SMD58
D7	SMD59
B5	SMD60
E6	SMD61
A4	SMD62

Table 93. Signal Name List (Signal Order) (Sheet 11 of 17)

Ball #	Signal Name
D4	SMD63
F18	SMDQM0
D18	SMDQM1
D13	SMDQM2
D12	SMDQM3
E18	SMDQM4
F17	SMDQM5
F14	SMDQM6
F13	SMDQM7
A24	SMOCLK
C20	SMRAS
C24	SMRCLK
A21	SMWE
AB23	SPARE0
AC23	SPARE1
N27	Unused
AB7	VCC
H23	VCC
H7	VCC
K23	VCC
K7	VCC
L7	VCC
M14	VCC
M15	VCC
M16	VCC
M24	VCC
N6	VCC
P12	VCC
P18	VCC
P24	VCC
R12	VCC
R18	VCC
T12	VCC
T18	VCC
T24	VCC
T6	VCC
V24	VCC

Table 93. Signal Name List (Signal Order) (Sheet 12 of 17)

Ball #	Signal Name
W7	VCC
Y23	VCC
Y7	VCC
AC8	VCC 18
G7	VCCA CPLL
AF26	VCCA DAC
AG27	VCCA DAC
AC20	VCCA DPLL0
F25	VCCA DPLL1
AE6	VCCA HPLL
AC10	VCCLM
AC11	VCCLM
AD11	VCCLM
AD12	VCCLM
AD13	VCCLM
AD17	VCCLM
AD18	VCCLM
AD19	VCCLM
AE18	VCCLM
AA23	VCCP 15G
AA26	VCCP 15G
J26	VCCP 15G
L23	VCCP 15G
M26	VCCP 15G
R26	VCCP 15G
U24	VCCP 15G
V26	VCCP 15G
AC9	VCCP CMOS
AE7	VCCP CMOS
AF6	VCCP CMOS
AC21	VCCP DVO
AF21	VCCP DVO
AF24	VCCP DVO
AD23	VCCP GPIO
AE25	VCCP GPIO
F26	VCCP HUB
J24	VCCP HUB

Table 93. Signal Name List (Signal Order) (Sheet 13 of 17)

Ball #	Signal Name
A12	VCCP SM
A8	VCCP SM
D11	VCCP SM
D14	VCCP SM
D17	VCCP SM
D20	VCCP SM
D23	VCCP SM
D26	VCCP SM
D5	VCCP SM
D8	VCCP SM
E12	VCCP SM
G10	VCCP SM
G11	VCCP SM
G23	VCCP SM
N24	VCCQ 15G
W23	VCCQ 15G
E15	VCCQ SM
F15	VCCQ SM
F7	VCCQ SM
G19	VCCQ SM
G20	VCCQ SM
AD16	VCCR
AE16	VCCR
V14	VCCR
V15	VCCR
V16	VCCR
AD15	VDD
AE15	VDD
B13	VSS
B3	VSS
B6	VSS
B9	VSS
B12	VSS
B15	VSS
B18	VSS
B21	VSS
B24	VSS

Table 93. Signal Name List (Signal Order) (Sheet 14 of 17)

Ball #	Signal Name
B27	VSS
A14	VSS
AB2	VSS
AB28	VSS
AC26	VSS
AC5	VSS
AC7	VSS
AD10	VSS
AD22	VSS
AD8	VSS
AD9	VSS
AE10	VSS
AE11	VSS
AE12	VSS
AE13	VSS
AE17	VSS
AE19	VSS
AE2	VSS
AE28	VSS
AE8	VSS
AE9	VSS
AF10	VSS
AF11	VSS
AF12	VSS
AF13	VSS
AF14	VSS
AF15	VSS
AF16	VSS
AF17	VSS
AF18	VSS
AF19	VSS
AF20	VSS
AF25	VSS
AF27	VSS
AF5	VSS
AF8	VSS
AF9	VSS

Table 93. Signal Name List (Signal Order) (Sheet 15 of 17)

Ball #	Signal Name
AG15	VSS
AG16	VSS
AG21	VSS
AG7	VSS
AH11	VSS
AH12	VSS
AH14	VSS
AH17	VSS
AH18	VSS
AH19	VSS
AH2	VSS
AH20	VSS
AH21	VSS
AH24	VSS
AH6	VSS
AH8	VSS
AH9	VSS
AJ21	VSS
AJ5	VSS
C21	VSS
C3	VSS
D2	VSS
D28	VSS
E10	VSS
E13	VSS
E14	VSS
E16	VSS
E19	VSS
E22	VSS
E25	VSS
E4	VSS
E7	VSS
F19	VSS
G2	VSS
G21	VSS
G28	VSS
G9	VSS

Table 93. Signal Name List (Signal Order) (Sheet 16 of 17)

Ball #	Signal Name
H25	VSS
H5	VSS
K2	VSS
K28	VSS
L25	VSS
L5	VSS
M12	VSS
M13	VSS
M17	VSS
M18	VSS
N12	VSS
N13	VSS
N14	VSS
N15	VSS
N16	VSS
N17	VSS
N18	VSS
N2	VSS
N28	VSS
P13	VSS
P14	VSS
P15	VSS
P16	VSS
P17	VSS
P25	VSS
P5	VSS
R13	VSS
R14	VSS
R15	VSS
R16	VSS
R17	VSS
T13	VSS
T14	VSS
T15	VSS
T16	VSS
T17	VSS
T2	VSS

Table 93. Signal Name List (Signal Order) (Sheet 17 of 17)

Ball #	Signal Name
T28	VSS
U12	VSS
U13	VSS
U14	VSS
U15	VSS
U16	VSS
U17	VSS
U18	VSS
U25	VSS
U5	VSS
V12	VSS
V13	VSS
V17	VSS
V18	VSS
W2	VSS
W28	VSS
Y25	VSS
Y5	VSS
G8	VSSA CPLL
AH26	VSSA DAC
AE20	VSSA DPLL0
G24	VSSA DPLL1
AD7	VSSA HPLL
AE29	VSYNC
AA5	VTT
AD5	VTT
AG5	VTT
E2	VTT
F5	VTT
J5	VTT
M5	VTT
R5	VTT
V5	VTT

Table 94. Signal Name List (Ball Order) (Sheet 1 of 17)

Ball #	Signal Name
A3	SMCLK3
A4	SMD62
A5	SMD26
A6	SMD57
A7	SMD22
A8	VCCP SM
A9	SMCKE3
A10	SMD19
A11	SMD49
A12	VCCP SM
A13	SMCKE0
A14	VSS
A15	SMCLK0
A16	SMA10
A17	SMA9
A18	SMA4
A19	SMA5
A20	SMA0
A21	SMWE
A22	SMD47
A23	SMD44
A24	SMOCLK
A25	SMD41
A26	SMD38
A27	SMD4
B2	SMCLK1
B3	VSS
B4	SMD30
B5	SMD60
B6	VSS
B7	SMD55
B8	SMD53
B9	VSS
B10	SMD52
B11	SMD50
B12	VSS
B13	VSS

Table 94. Signal Name List (Ball Order) (Sheet 2 of 17)

Ball #	Signal Name
B14	SMCLK2
B15	VSS
B16	SMBA0
B17	SMA8
B18	VSS
B19	SMA2
B20	SMA1
B21	VSS
B22	SMD15
B23	SMD12
B24	VSS
B25	SMD9
B26	SMD5
B27	VSS
B28	SMD34
C1	H_ADSB
C2	H_RCOMP
C3	VSS
C4	SMD31
C5	SMD28
C6	SMD58
C7	SMD24
C8	SMD21
C9	SMCKE1
C10	SMD20
C11	SMD18
C12	SMD16
C13	SMCKE2
C14	SMA12
C15	SMA11
C16	SMBA1
C17	SMA6
C18	SMA7
C19	SMA3
C20	SMRAS
C21	VSS
C22	SMD13

Table 94. Signal Name List (Ball Order) (Sheet 3 of 17)

Ball #	Signal Name
C23	SMD10
C24	SMRCLK
C25	SMD7
C26	SMD36
C27	SMD3
C28	SMD33
C29	SMD1
D1	H_HITMB
D2	VSS
D3	H_HITB
D4	SMD63
D5	VCCP SM
D6	SMD29
D7	SMD59
D8	VCCP SM
D9	SMD54
D10	SMD17
D11	VCCP SM
D12	SMDQM3
D13	SMDQM2
D14	VCCP SM
D15	SMCS3
D16	SMCS2
D17	VCCP SM
D18	SMDQM1
D19	SMCAS
D20	VCCP SM
D21	SMD46
D22	SMD43
D23	VCCP SM
D24	SMD39
D25	SMD37
D26	VCCP SM
D27	SMD2
D28	VSS
D29	SMD0
E1	H_BNRB

Table 94. Signal Name List (Ball Order) (Sheet 4 of 17)

Ball #	Signal Name
E2	VTT
E3	H_A4
E4	VSS
E5	SM VREF1
E6	SMD61
E7	VSS
E8	SMD25
E9	SMD23
E10	VSS
E11	N/C
E12	VCCP SM
E13	VSS
E14	VSS
E15	VCCQ SM
E16	VSS
E17	SMCS0
E18	SMDQM4
E19	VSS
E20	N/C
E21	SMD14
E22	VSS
E23	SMD8
E24	SMD6
E25	VSS
E26	SMD35
E27	SMD32
E28	HLD7
E29	HLD6
F1	H_A8
F2	H_A9
F3	H_A11
F4	H_DRDYB
F5	VTT
F6	SM RCOMP
F7	VCCQ SM
F8	SMD27
F9	SMD56

Table 94. Signal Name List (Ball Order) (Sheet 5 of 17)

Ball #	Signal Name
F10	SMD51
F11	SMD48
F12	N/C
F13	SMDQM7
F14	SMDQM6
F15	VCCQ SM
F16	SMCS1
F17	SMDQM5
F18	SMDQM0
F19	VSS
F20	N/C
F21	SMD45
F22	SMD11
F23	SMD40
F24	SM VREF0
F25	VCCA DPLL1
F26	VCCP HUB
F27	HLD5
F28	HLSTBB
F29	HLD4
G1	H_A13
G2	VSS
G3	H_A5
G4	H_TRDYB
G5	H_DBSYB
G6	H_RS2B
G7	VCCA CPLL
G8	VSSA CPLL
G9	VSS
G10	VCCP SM
G11	VCCP SM
G19	VCCQ SM
G20	VCCQ SM
G21	VSS
G22	SMD42
G23	VCCP SM
G24	VSSA DPLL1

Table 94. Signal Name List (Ball Order) (Sheet 6 of 17)

Ball #	Signal Name
G25	HLD8
G26	HLD0
G27	HLD9
G28	VSS
G29	HLSTB
H1	H_A15
H2	H_A3
H3	H_A28
H4	H_RS1B
H5	VSS
H6	H_RS0B
H7	VCC
H23	VCC
H24	HLREF
H25	VSS
H26	HLD10
H27	HLD3
H28	HLD1
H29	HLD2
J1	H_A25
J2	H_A19
J3	H_A10
J4	H_DEFERB
J5	VTT
J6	H_LOCKB
J7	H_GTL REF0
J23	HL RCOMP
J24	VCCP HUB
J25	G_REF
J26	VCCP 15G
J27	RGBA_DATA2
J28	RGBA_VSYNC
J29	RGBA_HSYNC
K1	H_A22
K2	VSS
K3	H_A31
K4	H_REQ3B

Table 94. Signal Name List (Ball Order) (Sheet 7 of 17)

Ball #	Signal Name
K5	H_REQ2B
K6	H_REQ0B
K7	VCC
K23	VCC
K24	RGBA_RCOMP
K25	RGBA_DATA0
K26	RGBA_DATA1
K27	RGBA_DATA4
K28	VSS
K29	RGBA_DATA5
L1	H_A24
L2	H_A20
L3	H_A23
L4	H_BPRIB
L5	VSS
L6	H_REQ4B
L7	VCC
L23	VCCP 15G
L24	RGBA_DATA8
L25	VSS
L26	RGBA_DATA3
L27	RGBA_DATA7
L28	RGBA_CLKOUT1
L29	RGBA_CLKOUT0
M1	H_A30
M2	H_A18
M3	H_A29
M4	H_REQ1B
M5	VTT
M6	H_A7
M12	VSS
M13	VSS
M14	VCC
M15	VCC
M16	VCC
M17	VSS
M18	VSS

Table 94. Signal Name List (Ball Order) (Sheet 8 of 17)

Ball #	Signal Name
M24	VCC
M25	RGBA_CLKIN
M26	VCCP 15G
M27	RGBA_DATA11
M28	RGBA_DATA9
M29	RGBA_DATA6
N1	H_A26
N2	VSS
N3	H_D6
N4	H_A6
N5	H_A14
N6	VCC
N12	VSS
N13	VSS
N14	VSS
N15	VSS
N16	VSS
N17	VSS
N18	VSS
N24	VCCQ 15G
N25	RESERVED
N26	RESERVED
N27	Unused
N28	VSS
N29	RGBA_DATA10
P1	H_D1
P2	H_D15
P3	H_D9
P4	H_A16
P5	VSS
P6	H_A12
P12	VCC
P13	VSS
P14	VSS
P15	VSS
P16	VSS
P17	VSS

Table 94. Signal Name List (Ball Order) (Sheet 9 of 17)

Ball #	Signal Name
P18	VCC
P24	VCC
P25	VSS
P26	M_GPIO0
P27	M_GPIO3
P28	RESERVED
P29	RESERVED
R1	H_D5
R2	H_D17
R3	H_D10
R4	H_A21
R5	VTT
R6	H_CPU RSTB
R12	VCC
R13	VSS
R14	VSS
R15	VSS
R16	VSS
R17	VSS
R18	VCC
R24	RESERVED
R25	RESERVED
R26	VCCP 15G
R27	RESERVED
R28	M_GPIO1
R29	M_GPIO2
T1	H_D18
T2	VSS
T3	H_D14
T4	H_A17
T5	H_A27
T6	VCC
T12	VCC
T13	VSS
T14	VSS
T15	VSS
T16	VSS

Table 94. Signal Name List (Ball Order) (Sheet 10 of 17)

Ball #	Signal Name
T17	VSS
T18	VCC
T24	VCC
T25	RESERVED
T26	RGBA_ALPHA1
T27	RGBA_ALPHA0
T28	VSS
T29	RESERVED
U1	H_D11
U2	H_D3
U3	H_D20
U4	H_D0
U5	VSS
U6	H_D4
U12	VSS
U13	VSS
U14	VSS
U15	VSS
U16	VSS
U17	VSS
U18	VSS
U24	VCCP 15G
U25	VSS
U26	RESERVED
U27	RGBA_ALPHA2
U28	RESERVED
U29	RESERVED
V1	H_D24
V2	H_D30
V3	H_D16
V4	H_D8
V5	VTT
V6	H_D12
V12	VSS
V13	VSS
V14	VCCR
V15	VCCR

Table 94. Signal Name List (Ball Order) (Sheet 11 of 17)

Ball #	Signal Name
V16	VCCR
V17	VSS
V18	VSS
V24	VCC
V25	RESERVED
V26	VCCP 15G
V27	RGBA_ALPHA3
V28	RESERVED
V29	RESERVED
W1	H_D23
W2	VSS
W3	H_D19
W4	H_D13
W5	H_D7
W6	H_D2
W7	VCC
W23	VCCQ 15G
W24	RESERVED
W25	RESERVED
W26	RESERVED
W27	RESERVED
W28	VSS
W29	RESERVED
Y1	H_D25
Y2	H_D32
Y3	H_D31
Y4	H_D21
Y5	VSS
Y6	H_D26
Y7	VCC
Y23	VCC
Y24	RESERVED
Y25	VSS
Y26	RESERVED
Y27	RESERVED
Y28	RESERVED
Y29	RESERVED

Table 94. Signal Name List (Ball Order) (Sheet 12 of 17)

Ball #	Signal Name
AA1	H_D34
AA2	H_D38
AA3	H_D22
AA4	H_D33
AA5	VTT
AA6	H_D35
AA7	H_GTL REF1
AA23	VCCP 15G
AA24	RESERVED
AA25	RESERVED
AA26	VCCP 15G
AA27	RESERVED
AA28	RESERVED
AA29	RESERVED
AB1	H_D36
AB2	VSS
AB3	H_D39
AB4	H_D28
AB5	H_D29
AB6	H_D43
AB7	VCC
AB23	SPARE0
AB24	RSTB
AB25	RESERVED
AB26	RESERVED
AB27	RESERVED
AB28	VSS
AB29	RESERVED
AC1	H_D45
AC2	H_D42
AC3	H_D49
AC4	H_D37
AC5	VSS
AC6	H_D44
AC7	VSS
AC8	VCC 18
AC9	VCCP CMOS

Table 94. Signal Name List (Ball Order) (Sheet 13 of 17)

Ball #	Signal Name
AC10	VCCLM
AC11	VCCLM
AC19	DREF CLK
AC20	VCCA DPLL0
AC21	VCCP DVO
AC22	RESERVED
AC23	SPARE1
AC24	G_BUSY
AC25	I2C DATA
AC26	VSS
AC27	RESERVED
AC28	RESERVED
AC29	RESERVED
AD1	H_D41
AD2	H_D40
AD3	H_D27
AD4	H_D47
AD5	VTT
AD6	H_D48
AD7	VSSA HPLL
AD8	VSS
AD9	VSS
AD10	VSS
AD11	VCCLM
AD12	VCCLM
AD13	VCCLM
AD14	RAM REF
AD15	VDD
AD16	VCCR
AD17	VCCLM
AD18	VCCLM
AD19	VCCLM
AD20	RESERVED
AD21	RESERVED
AD22	VSS
AD23	VCCP GPIO
AD24	GB CLKOUT

Table 94. Signal Name List (Ball Order) (Sheet 14 of 17)

Ball #	Signal Name
AD25	I2C CLK
AD26	DDC2 DATA
AD27	DDC1 DATA
AD28	HSYNC
AD29	RESERVED
AE1	H_D59
AE2	VSS
AE3	H_D52
AE4	H_D57
AE5	H_D51
AE6	VCCA HPLL
AE7	VCCP CMOS
AE8	VSS
AE9	VSS
AE10	VSS
AE11	VSS
AE12	VSS
AE13	VSS
AE14	RAM REF
AE15	VDD
AE16	VCCR
AE17	VSS
AE18	VCCLM
AE19	VSS
AE20	VSSA DPLL0
AE21	RESERVED
AE22	RESERVED
AE23	RESERVED
AE24	RESERVED
AE25	VCCP GPIO
AE26	DDC2 CLK
AE27	DDC1 CLK
AE28	VSS
AE29	VSYNC
AF1	H_D63
AF2	H_D55
AF3	H_D46

Table 94. Signal Name List (Ball Order) (Sheet 15 of 17)

Ball #	Signal Name
AF4	H_D54
AF5	VSS
AF6	VCCP CMOS
AF7	SCK
AF8	VSS
AF9	VSS
AF10	VSS
AF11	VSS
AF12	VSS
AF13	VSS
AF14	VSS
AF15	VSS
AF16	VSS
AF17	VSS
AF18	VSS
AF19	VSS
AF20	VSS
AF21	VCCP DVO
AF22	RESERVED
AF23	RESERVED
AF24	VCCP DVO
AF25	VSS
AF26	VCCA DAC
AF27	VSS
AF28	REDW
AF29	RED
AG1	H_D58
AG2	H_D53
AG3	H_D62
AG4	H_D60
AG5	VTT
AG6	GCLK
AG7	VSS
AG8	DQB7
AG9	DQB5
AG10	DQB3
AG11	RQ0

Table 94. Signal Name List (Ball Order) (Sheet 16 of 17)

Ball #	Signal Name
AG12	RQ2
AG13	RQ4
AG14	RQ6
AG15	VSS
AG16	VSS
AG17	DQA0
AG18	DQA2
AG19	DQA4
AG20	DQA6
AG21	VSS
AG22	RESERVED
AG23	RESERVED
AG24	RESERVED
AG25	RESERVED
AG26	GB CLKIN0
AG27	VCCA DAC
AG28	GREENW
AG29	GREEN
AH2	VSS
AH3	H_D50
AH4	H_D61
AH5	HTCLKB
AH6	VSS
AH7	CMD
AH8	VSS
AH9	VSS
AH10	DQB1
AH11	VSS
AH12	VSS
AH13	RQ3
AH14	VSS
AH15	CTM
AH16	CFM_B
AH17	VSS
AH18	VSS
AH19	VSS
AH20	VSS

Table 94. Signal Name List (Ball Order) (Sheet 17 of 17)

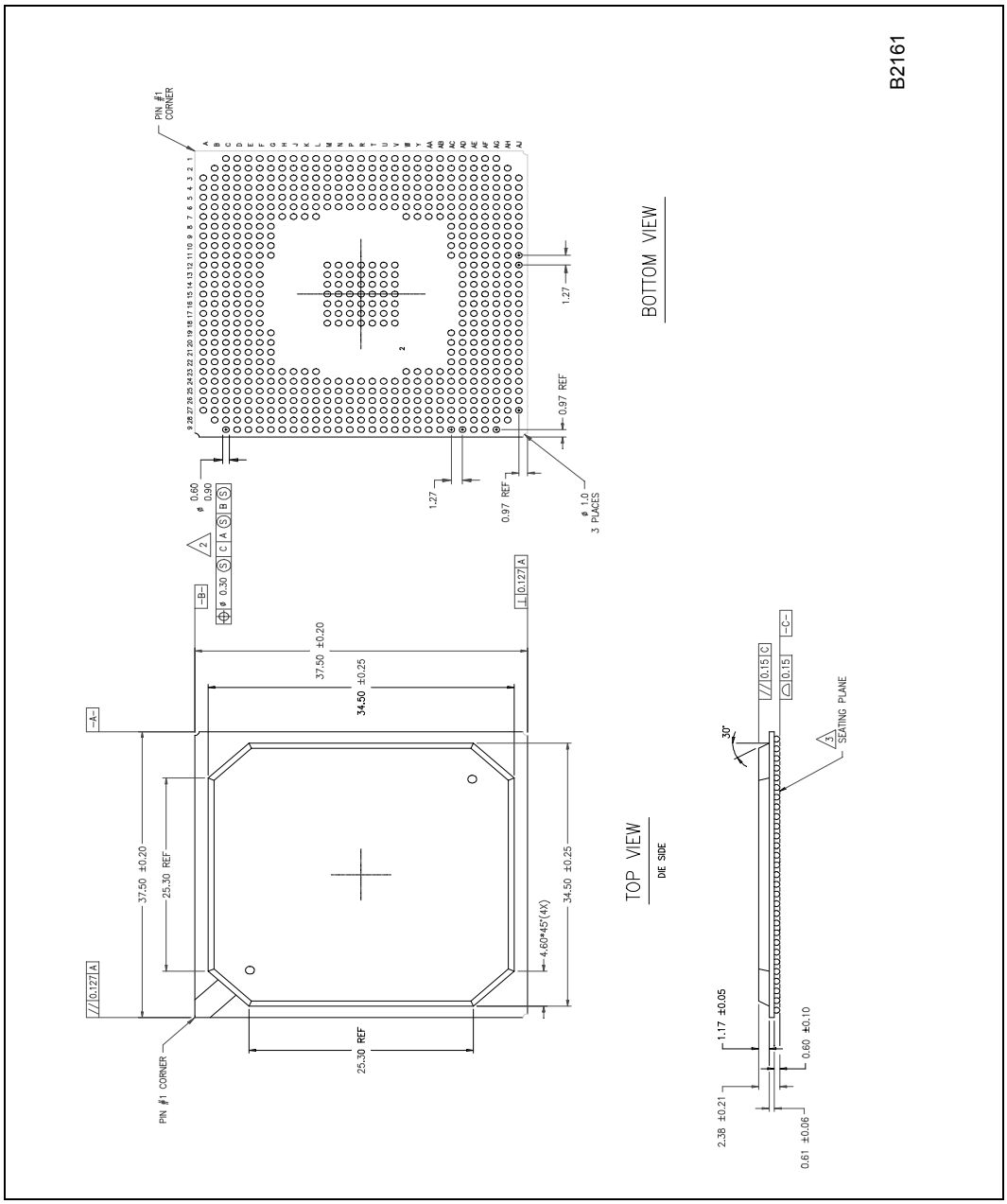
Ball #	Signal Name
AH21	VSS
AH22	RESERVED
AH23	RESERVED
AH24	VSS
AH25	RESERVED
AH26	VSSA DAC
AH27	BLUEW
AH28	BLUE
AJ3	H_D56
AJ4	HTCLK
AJ5	VSS
AJ6	RCLK
AJ7	SIO
AJ8	DQB6
AJ9	DQB4
AJ10	DQB2
AJ11	DQB0
AJ12	RQ1
AJ13	RQ5
AJ14	RQ7
AJ15	CTM_B
AJ16	CFM
AJ17	DQA1
AJ18	DQA3
AJ19	DQA5
AJ20	DQA7
AJ21	VSS
AJ22	RESERVED
AJ23	RESERVED
AJ24	RESERVED
AJ25	RESERVED
AJ26	RESERVED
AJ27	IREF

8.0 Package Dimensions and Marks

8.1 Package Dimensions

Figure 15 outlines the mechanical dimensions for the Intel® 835 chipset GMCH. The package is a 625-ball grid array (BGA) package.

Figure 15. Intel® 835 Chipset GMCH Package Dimensions



8.2 Package Mark Diagrams

Figure 16. Mark Diagram